CURSO PRÁTICO 60 DE PROGRAMAÇÃO DE COMPUTADORES

INPUT

PROGRAMAÇÃO BASIC – PROGRAMAÇÃO DE JOGOS – CÓDIGO DE MÁQUINA

NOVA CULTURAL

Cz$ 50,00

JÁ NAS BANCAS
A CAPA PARA ENCADERNAR
O VOLUME 4

INPUT

Vol. 4 N.º 60

NESTE NÚMERO

PROGRAMAÇÃO BASIC

MODELOS E SIMULAÇÃO

Balanço de uma empresa comercial. Variáveis de distribuição normal. Previsões meteorológicas. Condições de mercado. Controle de caixa 1181

CÓDIGO DE MÁQUINA

AVALANCHE: MAIS SALTOS

Saindo do chão. Salto à frente. Terra firme. Checagem da segurança. Avançando posições. Alterações do escore. Ajustes finais 1186

PROGRAMAÇÃO BASIC

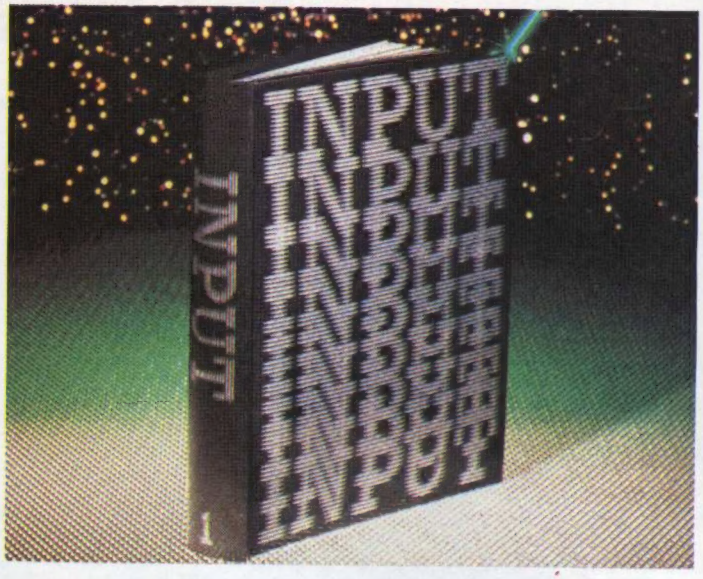
FIGURAS TRIDIMENSIONAIS

Desenhos em três dimensões. Criação das figuras. Construção do perfil. Como alterar o ângulo de visão. Rotação da figura 1194

PERIFÉRICOS

VIDEOTEXTO E MICROCOMPUTADORES

O que é o videotexto. Informações e serviços. Conexão ao microcomputador 1200



PLANO DA OBRA

"INPUT" é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: 1. PESSOALMENTE — Por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em **São Paulo**, os endereços são: rua Brigadeiro Tobias, 773, Centro; avenida Industrial, 117, Santo André; e no **Rio de Janeiro**: avenida Mem de Sá, 191/193, Centro. 2. POR CARTA — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidora Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132, Jardim Teresa — CEP 06000 — Osasco — SP. Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na agência do Correio. 3. POR TELEX — Utilize o nº (011) 33 670 DNAP.

Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações, Lda. — Qta. Pau Varais, Azinhaga de Fetais — 2 685, Camarate — Lisboa; Apartado 57 — Telex 43 069 JARLIS P.

Atenção: Após seis meses do encerramento da coleção, os pedidos serão atendidos dependendo da disponibilidade do estoque.

Obs.: Quando pedir livros, mencione sempre título e/ou autor da obra, além do número da edição.

COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.

NOVA CULTURAL

Editor
VICTOR CIVITA

REDAÇÃO
Diretor Editorial: Carmo Chagas

Editores Executivos: Antonio José Filho, Berta Sztark Amar

Editor Chefe: Paulo de Almeida
Editora de Texto: Ana Lúcia B. de Lucena
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Dagmar Bastos Sampaio, Grace Alonso Arruda, Monica Lenardon Corradi
Secretária de Redação/ Coordenadora: Stefania Crema
Secretários de Redação: Beatriz Hagström, José Benedito de Oliveira Damião, Maria de Lourdes Carvalho, Marisa Soares de Andrade, Mauro de Queiroz

COLABORADORES

Consultor Editorial Responsável: Dr. Renato M. E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
Execução Editorial: DATAQUEST Assessoria em Informática Ltda., Campinas, SP
Tradução, adaptação, programação e redação: Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Marcos Huascar Velasco, Raul Neder Porrelli, Ricardo J. P. de Aquino Pereira
Coordenação Geral: Rejane Felizatti Sabbatini

COMERCIAL
Diretor Comercial: Roberto Martins Silveira
Gerente Comercial: Flávio Maculan
Gerente de Circulação: Denise Maria Mozol

PRODUÇÃO
Gerente de Produção: João Stungis
Coordenador de Impressão: Atílio Roberto Bonon

Preparador de Texto/Coordenador: Eliel Silveira Cunha
Preparadores de Texto: Alzira Moreira Braz, Ana Maria Dilguerian, Levon Yacubian, Luciano Tasca, Maria Teresa Galluzzi, Maria Teresa Martins Lopes, Paulo Felipe Mendrone
Revisor/Coordenador: José Maria de Assis
Revisoras: Conceição Aparecida Gabriel, Isabel Leite de Camargo, Ligia Aparecida Ricetto, Maria de Fátima Cardoso, Nair Lucia de Britto
Paste-up: Anastase Potaris, Balduino F. Leite, Edson Donato

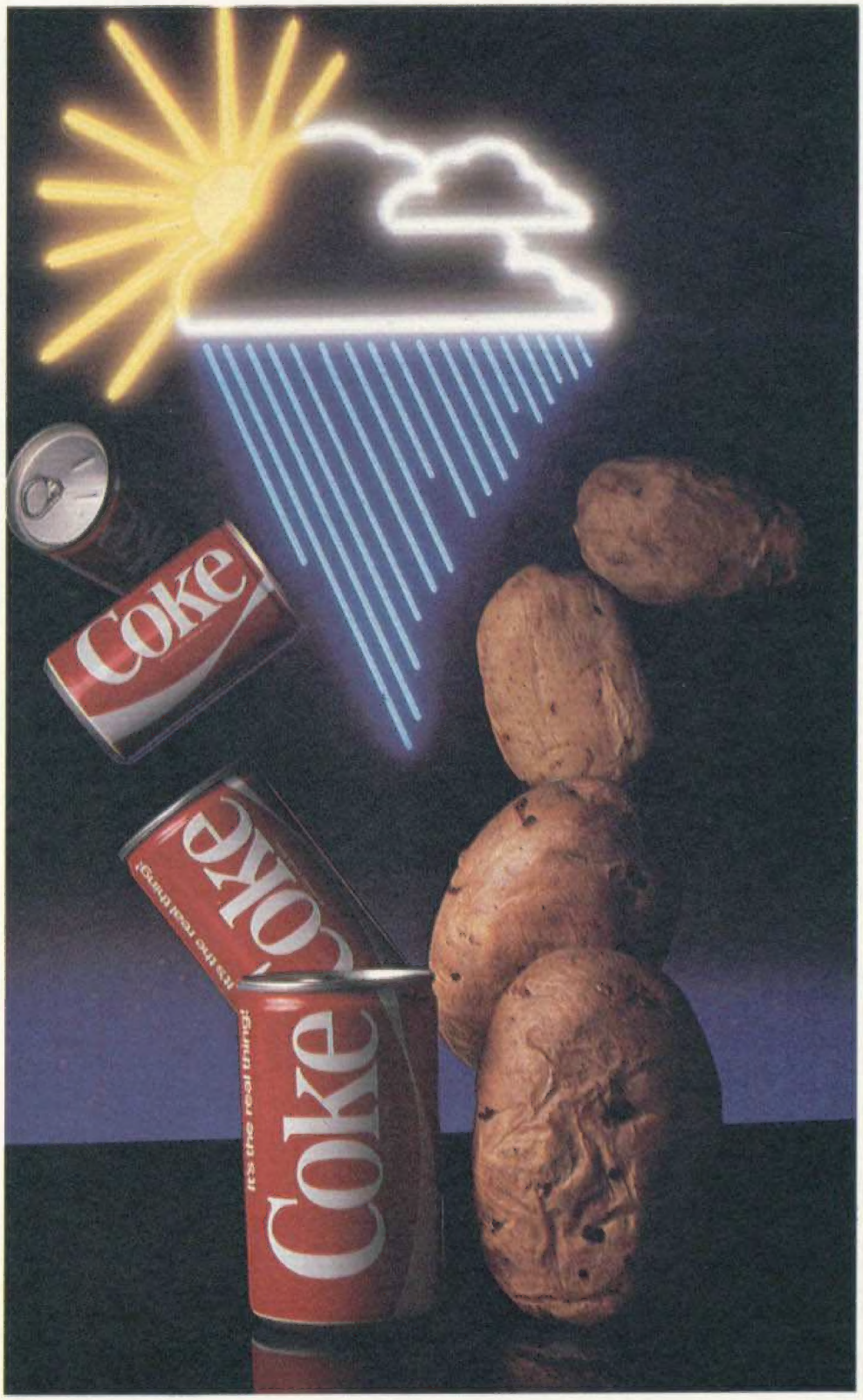
© Marshall Cavendish Limited 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986.

Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, n.º 2000 - 3º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta na AM Produções Gráficas Ltda. e impressa na Divisão Gráfica da Editora Abril S.A.

MODELOS E SIMULAÇÃO

- COMO SIMULAR O CLIMA
- VARIÁVEIS DE DISTRIBUIÇÃO NORMAL
- CONDIÇÕES DE MERCADO
- CONTROLE DE CAIXA

Retomando nossos estudos sobre o uso de modelos, apresentamos neste artigo um programa que se vale de alguns princípios econômicos para simular o balanço de uma empresa comercial.



No artigo da página 1176, examinamos o papel da simulação em várias áreas de pesquisa. Vimos também como gerar diferentes variáveis aleatórias, cada qual adequada a uma determinada situação. Como demonstrou nosso programa de simulação normal, a geração de variáveis normalmente distribuídas é muito simples, mas não tão eficiente — foram necessários quinze números aleatórios para criar uma única variável. O programa deste artigo usa um método mais eficaz, que permite simular diversos aspectos de uma pequena empresa comercial. Este programa também pode ser utilizado como um jogo (até bem divertido), mas constitui, de fato, um modelo de uma situação real.

```
 10 POKE 23658,8: POKE 23609,
 12
 20 PAPER 0: BORDER 0: INK 7:
 CLS
 30 PRINT AT 0,8; INVERSE 1;"
 BATATA QUENTE "
 40 DIM A$(4,12): DIM L(2):
 DIM D(10): DIM W(2): DIM Q(2)
 : DIM P(2)
 50 LET A$(1)="QUENTE E SECO":
 LET D(3)=150: LET D(4)=300
 60 LET A$(2)="QUENTE E UMIDO"
 : LET D(5)=100: LET D(6)=200
 70 LET A$(3)="FRIO E SECO":
 LET D(7)=250: LET D(8)=160
 80 LET A$(4)="FRIO E UMIDO":
 LET D(9)=200: LET D(10)=100
580 DIM C(4,2): LET C(1,1)=.1:
LET C(1,2)=.15
590 LET C(2,1)=0.5: LET C(2,2)
=0.25
600 LET C(3,1)=0.01: LET C(3,2
)=0.12
610 LET C(4,1)=10: LET C(4,2)=
10
620 INPUT "QUANTOS JOGADORES (
1-6) ? ";N
625 IF N<1 OR N>6 THEN  GOTO
620
```

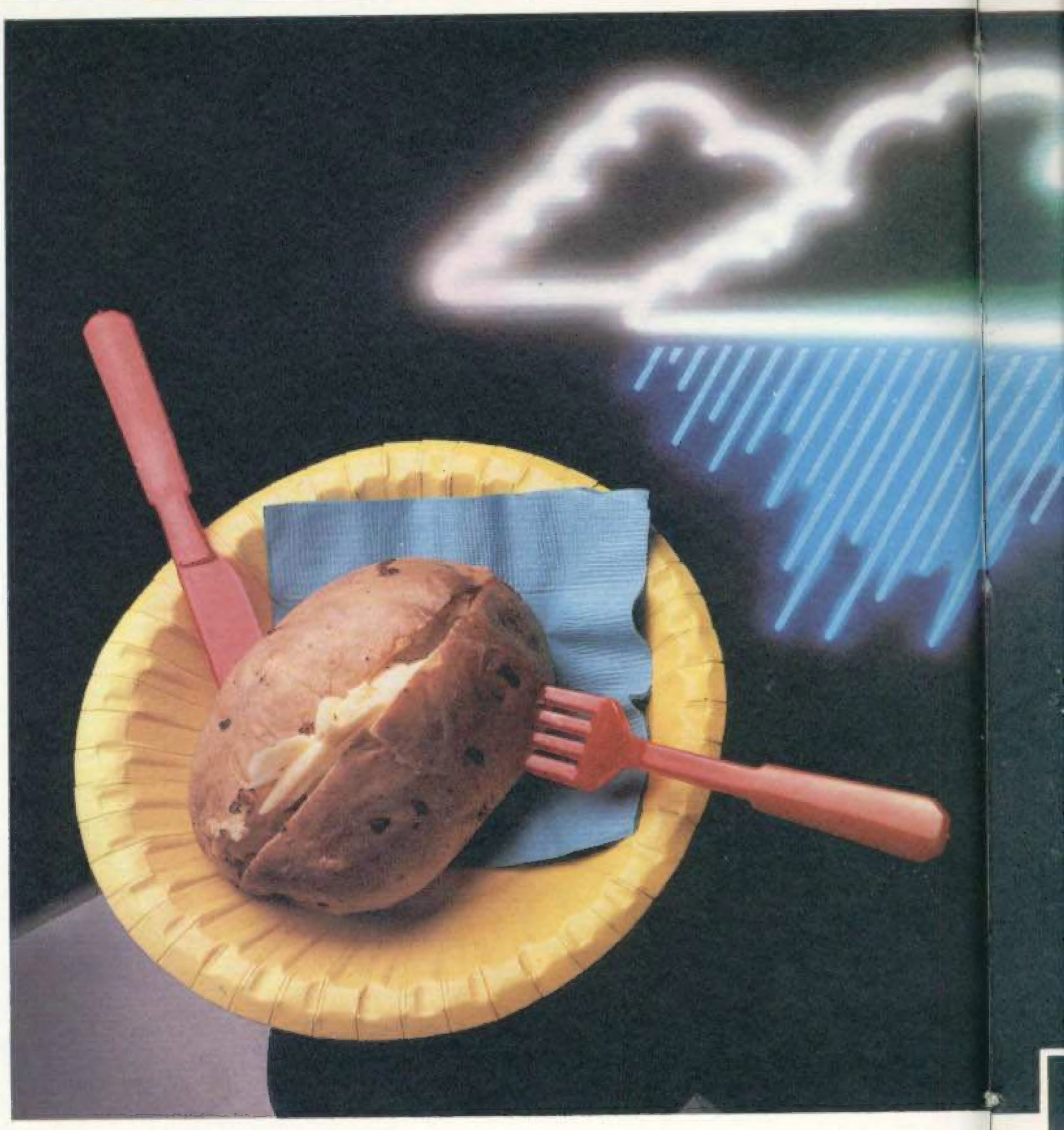
```
627 DIM K(N): DIM T(2,N): DIM
O(2,N)
630 FOR I=1 TO 2: FOR J=1 TO N
640 LET T(I,J)=0
650 NEXT J: NEXT I
700 FOR K=1 TO 10
710 LET P1=INT (10*(.3+(RND*1/
2)))/10
720 LET P2=INT (10*(.2+(RND*1/
2)))/10
730 PRINT  INK 4; BRIGHT 1;"--
---------------------------
 Dia:"; INK 7;K; INK 4;'"-----
---------------------------"''
740 PRINT "Prob. de um dia que
nte e seco : ";100*P1;"%"
750 PRINT "Prob. de um dia sec
o :          ";100*P2;"%"
760 GOSUB 1400
770 LET U1=RND*1: LET U2=RND*1
780 LET V1=SQR (2*(LN (1/U1)))
790 LET V2=COS (2*PI*U2): LET
V3=SIN (2*PI*U2)
800 LET Z1=INT (V1*V2): LET Z2
=INT (V1*V3)
810 LET A1=P1*P2: LET A2=P1
820 LET A3=P1+P2-A1: LET A4=1:
LET F=RND*1
821 PRINT : IF F<A1 THEN  LET
R=1
822 IF F>A1 AND F<=A2 THEN
LET R=2
823 IF F>A2 AND F<=A3 THEN
LET R=3
824 IF F>A3 AND F<=A4 THEN
LET R=4
830 CLS : PRINT "O TEMPO ESTA
";A$(R)
840 LET D(1)=INT (D(1+R*2)+Z1*
25): LET D(2)=INT (D(2+R*2)+Z2
*40)
850 PRINT "Demanda de batatas
assadas =";D(1)
860 PRINT "Demanda de latas de
coca =";D(2)
 990 PRINT ''
1000 PRINT  INK 5; INVERSE 1;"J
OGADOR  GANHO  CUSTOS  LUCRO  "
1010 GOSUB 1600
1020 NEXT K
1030 PAUSE 200
1090 CLS
1100 PRINT "  RESULTADO FINAL A
POS 10 DIAS "''
1110 PRINT "JOGADOR","LUCRO TOT
AL"
1120 FOR J=1 TO N
1130 PRINT J,K(J): NEXT J
1140 PRINT '': PRINT "FIM DE JO
GO"
1150 STOP
1400 PRINT "SEU PEDIDO POR FAVO
R ": PRINT
1410 FOR J=1 TO N
1420 PRINT "JOGADOR";J: PRINT
1430 INPUT "NUMERO DE BATATAS Q
UENTES ";O(1,J)
1440 INPUT "NUMERO DE LATAS DE
COCA COLA";O(2,J)
1450 NEXT J
1460 RETURN
1600 FOR J=1 TO N
1610 FOR I=1 TO 2
1620 LET L=O(I,J)
1630 IF D(I)<L THEN  LET L=D(I)
1650 LET W(I)=C(2,I)*L
1670 LET Q(I)=C(1,I)*O(I,J)
1680 IF D(I)>L(I) THEN  GOTO 17
00
1690 LET Q(I)=Q(I)-C(3,I)*(O(I,
J)-D(I))
1700 LET P(I)=W(I)-Q(I)
1710 LET T(I,J)=T(I,J)+P(I)
1720 NEXT I
1730 LET K(J)=T(1,J)+T(2,J)-200
1740 LET E=W(1)+W(2)
1750 LET C=Q(1)+Q(2)+20
1760 LET P=P(1)+P(2)-20
1770 PRINT  INK 6;TAB 3;J;TAB 9
;E;TAB 18;C;TAB 25;P;''
1780 NEXT J
1790 RETURN
```

```
5 R=RND(-TIME)
10 PI=4*ATN(1)
20 CLS
30 PRINT TAB(13);"Batata quente
":PRINT:PRINT
580 C1(1)=.1:C1(2)=.15
590 C2(1)=.5:C2(2)=.25
600 C3(1)=.01:C3(2)=.12
620 INPUT"QUANTOS JOGADORES (1-
6) ";N
625 IF N<1 OR N>6 THEN 620
630 FOR I=1 TO 2:FOR J=1 TO N
640 TP(I,J)=0
650 NEXT J,I
700 FOR K=1 TO 10
710 P1=INT(10*(.3+RND(1)/2))/10
720 P2=INT(10*(.2+RND(1)/2))/10
725 PRINT" QUALQUER TECLA PARA
CONTINUAR"
726 IF INKEY$="" THEN 726
730 PRINT"dia";K:PRINT
740 PRINT"PROB. DE UM DIA QUENT
E :";100*P1;"%"
750 PRINT"PROB. DE UM DIA SECO
:";100*P2;"%"
760 GOSUB 1400
770 U1=RND(1):U2=RND(1)
780 V1=SQR(2*(LOG(1/U1)))
790 V2=COS(2*PI*U2):V3=SIN(2*PI
*U2)
800 Z1=INT(V1*V2):Z2=INT(V1*V3)
810 A1=P1*P2:A2=P1
820 A3=P1+P2-A1:A4=1:F=RND(1)
```

```
821 CLS:IF F<=A1 THEN 830
822 IF F>A1 AND F<=A2 THEN 870
823 IF F>A2 AND F<=A3 THEN 810
824 IF F>A3 AND F<=A4 THEN 950
830 PRINT"O TEMPO ESTA QUENTE E
 SECO"
840 D(1)=150+Z1*25:D(2)=300+Z2*
40:GOTO 970
870 PRINT"O TEMPO ESTA QUENTE E
 UMIDO"
880 D(1)=100+Z1*25:D(2)=200+Z2*
40:GOTO 970
910 PRINT"O TEMPO ESTA FRIO E S
ECO"
920 D(1)=250+Z1*25:D(2)=160+Z2*
40:GOTO 970
950 PRINT"O TEMPO ESTA FRIO E U
MIDO"
960 D(1)=200+Z1*25:D(2)=100+Z2*
40
970 PRINT"DEMANDA DE BATATAS AS
SADAS=";D(1)
980 PRINT"DEMANDA DE LATAS DE C
OCA=";D(2)
1000 PRINT"JOGADOR  GANHO  CUST
OS  LUCRO"
1010 GOSUB 1600
1020 NEXT K
1030 FOR I=1 TO 2000:NEXT
1090 CLS
1100 PRINT" RESULTADO FINAL APO
S 10 DIAS ":PRINT:PRINT
1110 PRINT"JOGADOR","LUCRO TOTA
L"
1120 FOR J=1 TO N
1130 PRINT J,TT(J):NEXT J
1140 PRINT:PRINT:PRINT"FIM DE J
OGO"
1150 END
1400 PRINT:PRINTTAB(5);"seu ped
ido por favor":PRINT
1410 FOR J=1 TO N
1420 PRINT"JOGADOR";J:PRINT
1430 INPUT"NUMERO DE BATATAS";O
(1,J)
1440 INPUT"NUMERO DE LATAS DE C
OCA COLA";O(2,J)
1450 NEXT J
1460 RETURN
1600 FOR J=1 TO N
1610 FOR I=1 TO 2
1620 L=O(I,J)
1630 IF D(I)<L THEN L=D(I)
1650 RV(I)=C2(I)*L
1670 TC(I)=C1(I)*O(I,J)
1680 IF D(I)<=L THEN RC(I)=TC(I
)-C3(I)*(O(I,J)-D(I))
1700 P(I)=RV(I)-TC(I)
1710 TP(I,J)=TP(I,J)+P(I)
1720 NEXT I
1730 TT(J)=TP(1,J)+TP(2,J)-200
1740 E=RV(1)+RV(2)
1750 C=TC(1)+TC(2)+20
1760 P=P(1)+P(2)-20
1770 PRINTUSING"    $     ###.##
$###.## ###.##";J,E,C,P
1780 NEXT J
1790 RETURN
```

```
10 PI=4*ATN(1)
20 HOME
30 PRINT TAB(13);"BATATA QUENTE
":PRINT:PRINT
580 C1(1)=.1:C1(2)=.15
590 C2(1)=.5:C2(2)=.25
600 C3(1)=.01:C3(2)=.12
620 INPUT"QUANTOS JOGADORES (1-
6) ";N
625 IF N<1 OR N>6 THEN 620
630 FOR I=1 TO 2:FOR J=1 TO N
640 TP(I,J)=0
650 NEXT J,I
700 FOR K=1 TO 10
710 P1=INT(10*(.3+RND(1)/2))/10
720 P2=INT(10*(.2+RND(1)/2))/10
725 PRINT" QUALQUER TECLA PARA
CONTINUAR"
726 GET W$
730 PRINT"DIA";K:PRINT
740 PRINT"PROB. DE UM DIA QUENT
E :";100*P1;"%"
750 PRINT"PROB. DE UM DIA SECO
:";100*P2;"%"
760 GOSUB 1400
770 U1=RND(1):U2=RND(1)
780 V1=SQR(2*(LOG(1/U1)))
790 V2=COS(2*PI*U2):V3=SIN(2*PI
*U2)
800 Z1=INT(V1*V2):Z2=INT(V1*V3)
810 A1=P1*P2:A2=P1
820 A3=P1+P2-A1:A4=1:F=RND(1)
821 HOME:IF F<=A1 THEN 830
822 IF F>A1 AND F<=A2 THEN 870
823 IF F>A2 AND F<=A3 THEN 810
824 IF F>A3 AND F<=A4 THEN 950
830 PRINT"O TEMPO ESTA QUENTE E
 SECO"
840 D(1)=150+Z1*25:D(2)=300+Z2*
40:GOTO 970
870 PRINT"O TEMPO ESTA QUENTE E
 UMIDO"
880 D(1)=100+Z1*25:D(2)=200+Z2*
40:GOTO 970
910 PRINT"O TEMPO ESTA FRIO E S
ECO"
920 D(1)=250+Z1*25:D(2)=160+Z2*
40:GOTO 970
950 PRINT"O TEMPO ESTA FRIO E U
MIDO"
960 D(1)=200+Z1*25:D(2)=100+Z2*
40
970 PRINT"DEMANDA DE BATATAS AS
SADAS=";D(1)
980 PRINT"DEMANDA DE LATAS DE C
OCA=";D(2)
1000 PRINT"JOGADOR  GANHO  CUST
OS  LUCRO"
1010 GOSUB 1600
1020 NEXT K
1030 FOR I=1 TO 2000:NEXT
1090 HOME
1100 PRINT" RESULTADO FINAL APO
S 10 DIAS ":PRINT:PRINT
1110 PRINT"JOGADOR","LUCRO TOTA
L"
1120 FOR J=1 TO N
1130 PRINT J,TT(J):NEXT J
1140 PRINT:PRINT:PRINT"FIM DE J
OGO"
1150 END
1400 PRINT:PRINTTAB(5);"SEU PED
IDO POR FAVOR":PRINT
1410 FOR J=1 TO N
1420 PRINT"JOGADOR";J:PRINT
1430 INPUT"NUMERO DE BATATAS";O
(1,J)
1440 INPUT"NUMERO DE LATAS DE C
OCA COLA";O(2,J)
1450 NEXT J
1460 RETURN
1600 FOR J=1 TO N
1610 FOR I=1 TO 2
1620 L=O(I,J)
1630 IF D(I)<L THEN L=D(I)
1650 RV(I)=C2(I)*L
1670 TC(I)=C1(I)*O(I,J)
1680 IF D(I)<=L THEN RC(I)=TC(I
)-C3(I)*(O(I,J)-D(I))
1700 P(I)=RV(I)-TC(I)
1710 TP(I,J)=TP(I,J)+P(I)
1720 NEXT I
1730 TT(J)=TP(1,J)+TP(2,J)-200
1740 E=RV(1)+RV(2)
1750 C=TC(1)+TC(2)+20
1760 P=P(1)+P(2)-20
1770 PRINT TAB(2);J;TAB(9);E;TA
B(17);C;TAB(25);P
1780 NEXT J
1790 RETURN
```

```
10 PI=4*ATN(1)
20 CLS
30 PRINT @10,"batata quente":PR
INT:PRINT
580 C1(1)=.1:C1(2)=.15
590 C2(1)=.5:C2(2)=.25
600 C3(1)=.01:C3(2)=.12
620 INPUT"QUANTOS JOGADORES (1-
6) ";N
625 IF N<1 OR N>6 THEN 620
630 FOR I=1 TO 2:FOR J=1 TO N
640 TP(I,J)=0
650 NEXT J,I
700 FOR K=1 TO 10
710 P1=INT(10*(.3+RND(0)/2))/10
720 P2=INT(10*(.2=RND(0)/2))/10
725 PRINT" QUALQUER TECLA PARA
CONTINUAR"
726 IF INKEY$="" THEN 726
730 PRINT"dia";K:PRINT
740 PRINT"PROB. DE UM DIA QUENT
E :";100*P1;"%"
750 PRINT"PROB. DE UM DIA SECO
:";100*P2;"%"
760 GOSUB 1400
770 U1=RND(0):U2=RND(0)
780 V1=SQR(2*(LOG(1/U1)))
790 V2=COS(2*PI*U2):V3=SIN(2*PI
*U2)
800 Z1=INT(V1*V2):Z2=INT(V1*V3)
810 A1=P1*P2:A2=P1
820 A3=P1+P2-A1:A4=1:F=RND(0)
821 CLS:IF F<=A1 THEN 830
822 IF F>A1 AND F<=A2 THEN 870
823 IF F>A2 AND F<=A3 THEN 810
824 IF F>A3 AND F<=A4 THEN 950
830 PRINT"O TEMPO ESTA QUENTE E
 SECO"
840 D(1)=150+Z1*25:D(2)=300+Z2*
40:GOTO 970
870 PRINT"O TEMPO ESTA QUENTE E
 UMIDO"
880 D(1)=100+Z1*25:D(2)=200+Z2*
40:GOTO 970
910 PRINT"O TEMPO ESTA FRIO E S
ECO"
920 D(1)=250+Z1*25:D(2)=160+Z2*
40:GOTO 970
950 PRINT"O TEMPO ESTA FRIO E U
MIDO"
960 D(1)=200+Z1*25:D(2)=100+Z2*
40
970 PRINT"DEMANDA DE BATATAS AS
SADAS=";D(1)
980 PRINT"DEMANDA DE LATAS DE C
OCA=";D(2)
1000 PRINT"JOGADOR  GANHO  CUST
OS  LUCRO"
1010 GOSUB 1600
1020 NEXT K
1030 FOR I=1 TO 2000:NEXT
1090 CLS
1100 PRINT" RESULTADO FINAL APO
S 10 DIAS ":PRINT:PRINT
1110 PRINT"JOGADOR","LUCRO TOTA
L"
1120 FOR J=1 TO N
1130 PRINT J,TT(J):NEXT J
1140 PRINT:PRINT:PRINT"FIM DE J
OGO"
1150 END
```



```
1400 PRINT:PRINTTAB(5);"seu ped
ido por favor":PRINT
1410 FOR J=1 TO N
1420 PRINT"JOGADOR";J:PRINT
1430 INPUT"NUMERO DE BATATAS";O
(1,J)
1440 INPUT"NUMERO DE LATAS DE C
OCA COLA";O(2,J)
1450 NEXT J
1460 RETURN
1600 FOR J=1 TO N
1610 FOR I=1 TO 2
1620 L=O(I,J)
1630 IF D(I)<L THEN L=D(I)
1650 RV(I)=C2(I)*L
1670 TC(I)=C1(I)*O(I,J)
1680 IF D(I)<=L THEN RC(I)=TC(I
)-C3(I)*(O(I,J)-D(I))
1700 P(I)=RV(I)-TC(I)
1710 TP(I,J)=TP(I,J)+P(I)
1720 NEXT I
1730 TT(J)=TP(1,J)+TP(2,J)-200
1740 E=RV(1)+RV(2)
1750 C=TC(1)+TC(2)+20
1760 P=P(1)+P(2)-20
1770 PRINTUSING"   #     ###.##
####.## ###.##";J,E,C,P
1780 NEXT J
1790 RETURN
```

O programa focaliza os lucros e perdas de uma empresa do setor de alimentos, oferecendo ao usuário a opção de atuar sozinho ou de negociar com até cinco outros "comerciantes".

O usuário também pode jogar no lugar de outras pessoas, tomando decisões diferentes conforme o papel assumido — o que lhe dá a oportunidade de comparar os resultados que obtém, agindo cautelosamente ou se arriscando em lances ousados. Tendo feito sua escolha, ele deve fornecer o número de participantes ao programa.

ELEMENTO DE ACASO

Cada jogador administra um bar que vende batatas assadas e latas de refrigerante. A demanda por um ou por outro produto depende do clima. Se está fa-

zendo frio, aumenta a procura pelas batatas; se faz calor, compra-se mais refrigerante. Infelizmente, o gerente precisa adquirir o estoque na véspera, desconhecendo, portanto, o tempo que vai fazer. Pode contar, até certo ponto, com o serviço de meteorologia, que costuma acertar, em média, 70% de suas previsões. Após dez dias de compras e vendas, o administrador que obtiver maior lucro vence.

PREVISÕES

A primeira parte do programa (até a linha 650) cuida das variáveis que controlam a tela e as condições de compra e venda. Você paga um aluguel de $ 20 por dia. As batatas custam $ 0,10 cada e são vendidas a $ 0,50. O refrigerante custa $ 0,15 para compra e $ 0,25 para venda. As sobras do estoque de um dia para outro são vendidas a um preço mais baixo — $ 0,01, as batatas e $ 0,12, o refrigerante. Cada partida dura dez dias.

O indicador da demanda de cada produto, conforme o clima, está na tabela da página 1185. Aqui entra um elemento importante do modelo. Naturalmente, os dias melhores para a venda de batatas assadas são os piores para a venda de refrigerantes e vice-versa. Contudo, há uma demanda suficiente para manter as vendas de ambos os produtos. Tudo depende de como aproveitamos as informações fornecidas pelo serviço de meteorologia, sempre tendo em mente, porém, que as previsões não são totalmente confiáveis. As probabilidades para um dia quente e seco são determinadas nas linhas 710 e 720, e depois utilizadas para simular as condições do tempo (linhas 810 a 824).

VARIÁVEIS ALEATÓRIAS

As linhas 770 a 800 contêm um método sofisticado para gerar variáveis aleatórias de distribuição normal. Elas são empregadas para simular a demanda na linha 840. A linha 770 cria duas variáveis aleatórias (**U1** e **U2**), processadas por meio de três fórmulas matemáticas. Na linha 780, **U1** é invertida e depois elevada ao quadrado; em seguida, calculamos o seu logaritmo natural e a raiz quadrada deste **V1**. A linha 790 faz **V2** igual ao co-seno de uma circunferência de raio **U2**, e **V3** igual ao seno dessa circunferência. **V1**, **V2** e **V3** sofrem um processamento adicional na linha 800 para resultar nas variáveis normais **Z1** e **Z2**.

As linhas restantes do programa cuidam da entrada e da impressão dos resultados na tela.

Para ser bem-sucedido neste jogo é preciso cuidar de cada centavo. Os gastos podem ser assustadoramente próximos dos ganhos — mesmo após dez dias de atividade.

INDICADOR DE DEMANDA

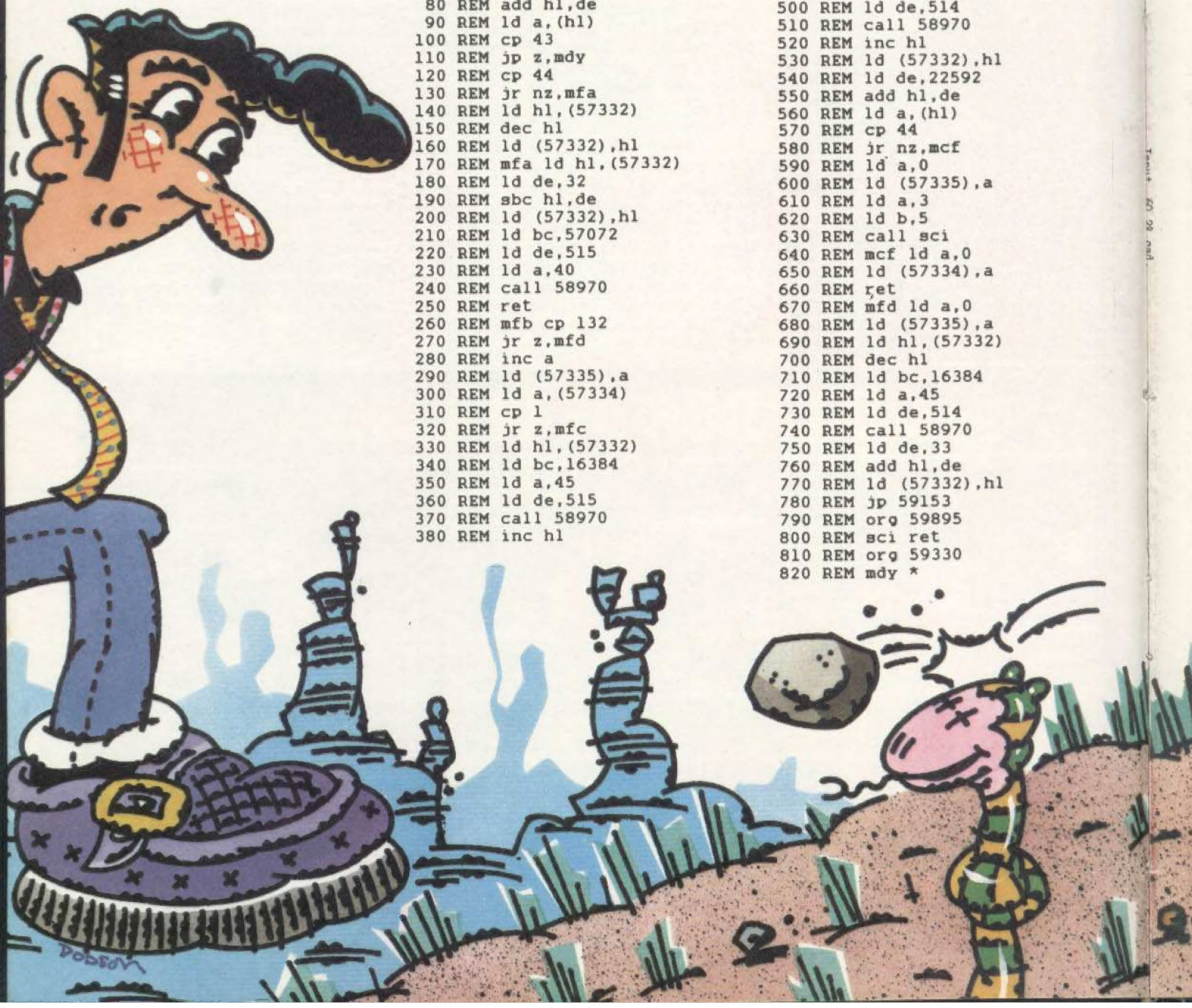
Clima	Intervalo de demanda	
Quente e seco	100-200	220-380
Quente e úmido	50-150	120-280
Frio e seco	200-300	80-240
Frio e úmido	150-250	20-180

AVALANCHE: MAIS SALTOS

Até agora, Willie só podia pular verticalmente. A rotina deste artigo vai torná-lo capaz de saltar à frente, ultrapassando os obstáculos como um ágil cabrito montês.

Esta é a parte final da série de rotinas que cuidam dos movimentos de Willie. Finalmente, veremos nosso personagem não só subir a encosta e se desviar das pedras, como também saltar sobre os buracos e as cobras.

```
 10 REM org 59472
 20 REM mfj cp 129
 30 REM jr nz,mfb
 40 REM inc a
 50 REM ld (57335),a
 60 REM ld hl,(57332)
 70 REM ld de,22561
 80 REM add hl,de
 90 REM ld a,(hl)
100 REM cp 43
110 REM jp z,mdy
120 REM cp 44
130 REM jr nz,mfa
140 REM ld hl,(57332)
150 REM dec hl
160 REM ld (57332),hl
170 REM mfa ld hl,(57332)
180 REM ld de,32
190 REM sbc hl,de
200 REM ld (57332),hl
210 REM ld bc,57072
220 REM ld de,515
230 REM ld a,40
240 REM call 58970
250 REM ret
260 REM mfb cp 132
270 REM jr z,mfd
280 REM inc a
290 REM ld (57335),a
300 REM ld a,(57334)
310 REM cp 1
320 REM jr z,mfc
330 REM ld hl,(57332)
340 REM ld bc,16384
350 REM ld a,45
360 REM ld de,515
370 REM call 58970
380 REM inc hl
390 REM ld (57332),hl
400 REM ld bc,57000
410 REM ld a,40
420 REM ld de,258
430 REM call 58970
440 REM ld a,1
450 REM ld(57334),a
460 REM ret
470 REM mfc ld hl,(57332)
480 REM ld bc,57016
490 REM ld a,40
500 REM ld de,514
510 REM call 58970
520 REM inc hl
530 REM ld (57332),hl
540 REM ld de,22592
550 REM add hl,de
560 REM ld a,(hl)
570 REM cp 44
580 REM jr nz,mcf
590 REM ld a,0
600 REM ld (57335),a
610 REM ld a,3
620 REM ld b,5
630 REM call sci
640 REM mcf ld a,0
650 REM ld (57334),a
660 REM ret
670 REM mfd ld a,0
680 REM ld (57335),a
690 REM ld hl,(57332)
700 REM dec hl
710 REM ld bc,16384
720 REM ld a,45
730 REM ld de,514
740 REM call 58970
750 REM ld de,33
760 REM add hl,de
770 REM ld (57332),hl
780 REM jp 59153
790 REM org 59895
800 REM sci ret
810 REM org 59330
820 REM mdy *
```



- SAINDO DO CHÃO
- SALTO À FRENTE
- SUBIDA DA ENCOSTA
- TERRA FIRME
- CHECAGEM
- DA SEGURANÇA
- AVANÇANDO POSIÇÕES
- ALTERAÇÕES DO ESCORE
- AJUSTES FINAIS

Quando o programa chama esta rotina pela primeira vez, o acumulador contém 129. Esse valor é armazenado na posição 57335 da memória quando as teclas M e N são pressionadas juntas.

SAINDO DO CHÃO

Antes de mais nada, esta rotina verifica se o acumulador contém o valor 129. O número também poderia ser 130, 131 ou 132 — e, se qualquer um deles estiver presente, os comandos **cp 129** e **jr nz,mfb** desviam o programa para o rótulo **mfb**. Na primeira vez que se chama a rotina, porém, o número no acumulador é 129.

A tarefa seguinte consiste em incrementar o acumulador e carregá-lo de volta em 57335. A instrução **ld hl,(57332)** transfere para a posição de Willie de 57332 para o par HL. O par DE é carregado com 22561 e adicionado em HL, para que obtenhamos a cor da posição adiante dos pés de Willie. A instrução **ld a,(hl)** carrega no acumulador o código dessa cor. Em seguida, o código é comparado com 43, a cor da cobra. Se esse valor é encontrado, Willie está saltando sobre a língua da cobra, e, fatalmente, leva uma picada e morre. A instrução **jp z,mdy** desvia então a rotina para o rótulo **mdy**, que providencia a eliminação de Willie.

Se ele não morreu, o programa compara a cor da posição adiante de seus pés com 44, a cor da terra. Se não há essa cor na frente de Willie, a instrução **jr nz,mfa** desvia o programa para a rotina **mfa**. Caso contrário, a posição de Willie é diminuída de um.

Em **mfa**, a posição de Willie é colocada em HL. DE é carregado com 32 e subtraído de HL, para mover o apontador um caractere acima na tela. Esse valor é colocado de volta em 57332.

O par BC é carregado com 57072, o endereço inicial para os dados de uma das figuras de Willie no salto para a frente. O par DE é carregado com 515, para especificar que um bloco dois por três será impresso — 2 x 256 + 3 = 515. A cor de Willie — 40, azul sobre ciano — é carregada com A, e a rotina de impressão é chamada.

Na próxima vez que ela for chamada, o acumulador conterá 130. Logo, o processador pulará a primeira parte, executando-a a partir do rótulo **mfb**.

Essa pequena rotina começa com um

teste. O conteúdo do acumulador é comparado com 132. Se esse valor está presente, a rotina **mfd** é chamada. Portanto, caso o acumulador contenha 130 ou 131, a rotina acionada é a **mfb**.

A primeira coisa que **mfb** faz é incrementar A e carregar o resultado de volta em 57335.

Em seguida, a variável da caminhada é carregada da posição de memória 57334 para o acumulador. Com isso, informa-se ao processador qual das duas figuras de Willie é necessária.

O conteúdo do acumulador é comparado com 1. Se 1 está presente, o programa salta para o rótulo **mfc** e imprime a figura de Willie com as pernas abertas. Antes que ela seja impressa, qualquer vestígio da figura anterior deve ser apagada. Para isso, a posição de Willie é transferida de 57332 para o par HL. O par BC é carregado com 16384, o início da tela. A é carregado com 45, para selecionar a cor ciano sobre ciano, e o par DE, com 515, para selecionar um bloco dois por três — 2 x 256 + 3 = 515. A rotina de impressão de bloco em 58970 é chamada, apagando a figura anterior de Willie.

HL é incrementado, movendo o apontador de tela para a próxima posição. Esse valor é colocado de volta no apontador do endereço 57332, ajustando-o também. BC é carregado com 57000, que é o início dos dados para a figura de Willie com as pernas juntas. A é ajustado com 40 — cor azul sobre ciano. DE é ajustado com 258 — o bloco de um por dois caracteres.

Impressa a figura de Willie, A é carregado com 1. Esse valor é armazenado em 57334, para que, na próxima vez, a outra figura seja impressa.

WILLIE AVANÇA

Quando a rotina for chamada novamente, a variável da caminhada em 57334 conterá 1 e o processador irá diretamente para o rótulo **mfc**. Nessa rotina, o par HL é carregado mais uma vez com a posição de Willie. BC é carregado com 57016, o endereço inicial para os dados da figura do personagem com as pernas abertas. A é ajustado de novo com 40, e DE, com 515 — a figura de Willie com as pernas abertas ocupa um bloco de dois por dois. A rotina de impressão é chamada, e Willie é impresso com as pernas abertas.

Observe que HL não foi incrementado nesta parte da rotina porque Willie ocupa, agora, duas posições. Ele avançou, portanto, aproximadamente metade de um caractere.

Uma vez que a figura tenha sido impressa, o apontador de tela em HL e 57332 é incrementado, fazendo com que Willie efetivamente avance.

SUBIDA DA ENCOSTA

Em seguida, devemos verificar se Willie está pisando na encosta ou não. DE é carregado com 22592, que é adicionado à posição de tela em HL. Com isso, ele passa a apontar para a cor da posição abaixo dos pés de Willie. A instrução **ld a,(hl)** carrega essa cor no acumulador, onde é comparada com 44. Se a cor da encosta não se encontra em A, Willie está no espaço aberto — e a instrução **jr nz,mcf** manda o processador para o rótulo **mcf**.

Mas, se o acumulador contém a cor da encosta, Willie conseguiu subir um aclive, merecendo um número adicional de pontos. É preciso, então, interromper o salto e atualizar o escore. Para isso, 0 é carregado em A e armazenado no endereço 57335. O valor 3 é carregado em A e 5 em B. Esses parâmetros serão utilizados por outra rotina, **sci** — ou incremento do escore. Na verdade, adicionaremos o valor 50 ao escore. O número 5 será adicionado à coluna das dezenas, que é a terceira a partir da esquerda. Em seguida, a rotina **sci** é chamada — como ela ainda não foi digitada, colocamos um **ret** provisório em seu endereço inicial.

Uma vez marcados os pontos, A é carregado com 0 e armazenado em 57334, quer Willie esteja pisando na encosta ou não. Indica-se, assim, que a próxima figura a ser impressa é a do personagem com as pernas juntas.

TERRA FIRME

Quando todas as figuras de Willie saltando já tiverem sido exibidas na tela, a variável de salto em 57335 terá o valor 132. Portanto, na próxima vez que a rotina de movimentação for chamada, o programa irá direto para a última rotina que começa com o rótulo **mfd**. Essa rotina começa igualando a variável de salto a 0. O valor 0 é carregado também em A e armazenado em 57335. A seguir, a posição de Willie na tela é decrementada.

O par BC é carregado com 16384, endereço do início da tela. A é igualado a 45, o código de ciano sobre ciano, e o par DE traz 514 para um bloco de dois por dois.

Chamando a rotina de impressão de bloco, apagaremos Willie da tela. O par DE é carregado com 33 e adicionado em HL. Com isso, movemos o apontador uma linha para baixo, colocando Willie de novo em terra firme. O resultado é armazenado em 57332.

Finalmente, a rotina de movimentação é acessada pela instrução **jp**. Essa rotina imprimirá Willie novamente de pé em sua posição sobre o solo, e pesquisará o teclado.

```
10   ORG 20321
20   MFJ CMPA #129
30   BNE MFB
40   INC 18261
50   LDX 18249
60   LEAX 290,X
70   LDA ,X
80   CMPA #$57
90   LBEQ MDY
100   JSR MFZ
110   LDX 18249
120   LEAX -255,X
130   STX 18249
140   LDU #17814
150   JSR CHARPR
160   LEAX 254,X
170   LDU #17846
180   JSR CHARPR
190   RTS
200   MFB CMPA #130
210   BNE MFC
220   INC 18261
230   JSR MFZ
240   LDX 18249
250   LEAX -255,X
260   STX 18249
270   LDU #17870
280   JSR CHARPR
290   LEAX 254,X
300   JSR CHARPR
310   LEAX 254,X
320   JSR CHARPR
330   LDX 18249
340   LEAX 864,X
350   LDA ,X+
360   CMPA #$FF
370   BEQ MFF
380   LDA ,X
390   CMPA #$FF
400   BEQ MFF
410   RTS
420   MFF LDA #4
430   LDB #5
440   JSR SCI
450   RTS
460   MFC CMPA #131
470   BNE MFD
480   INC 18261
490   JSR MFZ
500   LEAX 254,X
510   LDU #1536
520   JSR CHARPR
530   LDX 18249
540   LEAX 257,X
550   STX 18249
560   LDU #17814
570   JSR CHARPR
580   LEAX 254,X
```

```
590   LDU #17846
600   JSR CHARPR
610   RTS
620   MFD CMPA #132
630   BNE MFE
640   JSR MFZ
650   LDX 18249
660   PSHS X
670   LEAX 512,X
680   LDY ,X
690   PULS X
700   CMPY #$5555
710   BNE MFE
720   LEAX 1,X
730   STX 18249
740   MFE CLR 18261
750   CLR 18251
760   LDX 18249
770   LDU #17774
780   JSR CHARPR
790   LEAX 254,X
800   JSR CHARPR
810   RTS
820   MFZ LDX 18249
830   LDU #1536
840   JSR CHARPR
850   LEAX 254,X
860   JSR CHARPR
870   RTS
880   CHARPR EQU 19402
890   SCI EQU 20751
900   MDY EQU 20126
```

Quando o processador executar esta rotina pela primeira vez, o acumulador conterá o número 129. Esse número é armazenado na posição de memória 18261 pela primeira parte da rotina de movimentação de Willie, quando M e N foram pressionadas simultaneamente.

PREPARAÇÃO DO SALTO

De início, a rotina verifica se o valor 129 se encontra no acumulador. Se em seu lugar estiverem os números 130, 131 ou 132, as instruções **CMPA #129** e **BNE MFB** desviam o programa para **MFB**. Na primeira vez que se chama esta rotina, porém, A contém 129.

O acumulador é incrementado e carregado de volta em 18261; em consequência, a segunda parte desta rotina será chamada na próxima vez.

A instrução **LDX 18249** transfere a posição de Willie de 18249 para X. O número 290 é adicionado para indicar o byte adiante dos pés do personagem. **LDA ,X** carrega o conteúdo desse byte no acumulador e a instrução **CMPA #$57** compara-o com $57, a cor gráfica para a língua da cobra. Se houver essa cor na frente de Willie no momento do salto, ele será picado e morrerá. Caberá à instrução **LBEQ MDY** comandar a execução da rotina da morte.

Se nosso personagem ainda está vivo, o processador vai para a rotina **MFZ**, que apaga os dois caracteres abaixo de Willie — para que os pés da figura anterior não permaneçam ali.

A posição de Willie, transferida outra vez de 18249 para X, é subtraída de 255, para mover a posição um caractere para cima. A nova posição na tela é armazenada de volta em 18249.

O registro U é então carregado com 17814, endereço inicial para os dados da figura com as pernas abertas. Lembre-se de que U é usado como apontador de dados pela rotina **CHARPR**. Essa rotina, que utiliza os dados como pilha do usuário, é chamada; X é somado a 254, movendo-se para a posição inicial da metade inferior de Willie. U é carregado com 17846, o início dos dados para a metade inferior da figura, e **CHARPR** é chamada mais uma vez, imprimindo Willie sobre a encosta.

SAINDO DO CHÃO

Quando a rotina de movimentação for chamada de novo, o acumulador conterá 130, e a rotina **MFB** será executada. Nas chamadas posteriores, o acumulador conterá um valor maior e as instruções **CMPA #130** e **BNE MFC** desviarão o fluxo de processamento para a próxima parte do programa. Como sempre, a primeira coisa que a rotina faz é incrementar o conteúdo de 18261. Depois, salta para **MFZ**, onde a metade inferior de Willie é apagada.

A posição seguinte do personagem é carregada em X e subtraída de 255 uma vez mais. Isso move o apontador de tela um caractere acima. O registrador U é carregado com 17870 e **CHARPR** é

chamada três vezes. Entre cada chamada, X é incrementado com 254. Imprime-se, assim, uma nova figura de Willie, em três caracteres. Embora sua altura permaneça a mesma, o personagem ocupa um caractere adicional porque foi deslocado meio caractere para cima.

SALVO DA PEDRA

Willie recebe mais pontos quando se desvia de uma pedra, pulando por cima dela. É necessário, portanto, verificar se há uma pedra sob a figura.

A posição de Willie é transferida de 18249 para X, sendo somada a 864, de modo a indicar a posição imediatamente abaixo. A é carregado com o valor do byte da tela apontado por X, e esse registrador é incrementado.

O byte da tela no registrador A é comparado com $FF, a cor gráfica da pedra. Se ela estiver presente, o processador vai para o rótulo **MFF**. Caso contrário, o próximo byte é carregado e comparado. Se a pedra estiver ali, o processador passa para o rótulo **MFF**; se não, ele retorna.

A rotina **MFF** atualiza os pontos. A é carregado com 4, para indicar o dígito que será ajustado — o quarto a partir da esquerda (dezenas). B é carregado com 5. A seguir, o programa salta para a rotina **SCI**, que tem a função de adicionar 50 pontos ao escore. Como ainda não digitamos essa rotina, convém colocar **RTS** em sua posição inicial, para evitar um erro no programa.

WILLIE AVANÇA

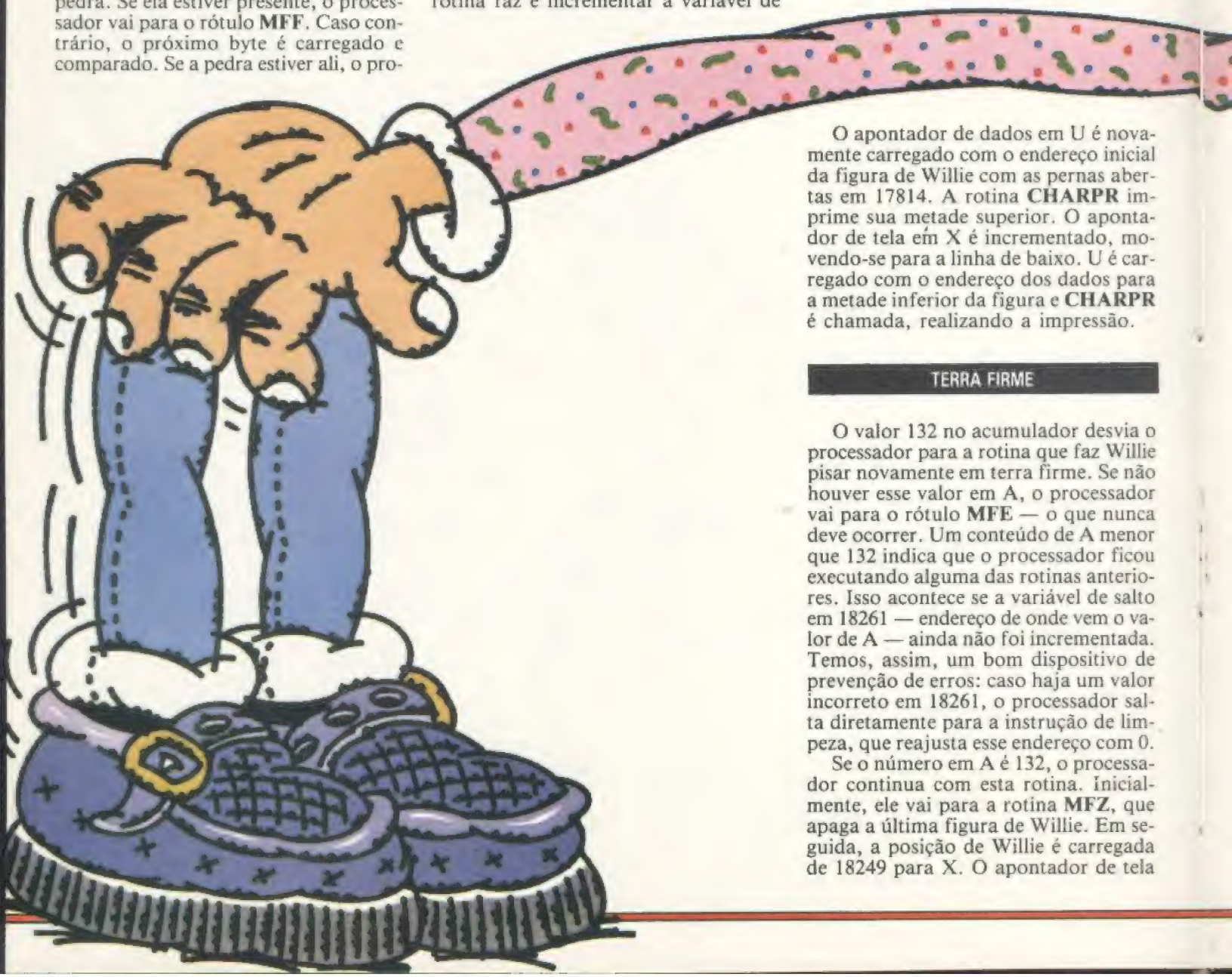
Na próxima vez que a rotina de movimentação é chamada, o acumulador contém 131. Caso esse valor não esteja presente, o programa é mandado para o rótulo **MFD**, pois o salto de Willie se encontra numa etapa mais adiantada.

Novamente, a primeira coisa que a rotina faz é incrementar a variável de salto em 18261. Quando a rotina voltar a ser chamada, o processador irá executar diretamente a parte seguinte.

Na seqüência, o programa salta para a rotina **MFZ**, que apaga os caracteres superior e intermediário de Willie. Seus pés serão apagados com um caractere do céu pela próxima rotina. Esse trabalho de limpeza adicional precisou ser feito porque Willie tinha três caracteres de altura na última vez.

A posição em X, que foi ajustada durante a execução da rotina **MFZ**, é incrementada com 254, movendo o apontador para os pés de Willie. O apontador de dados em U é carregado com 1536, o endereço do céu limpo, no canto superior esquerdo da tela. **CHARPR** é chamada e imprime o caractere de céu.

A posição anterior de Willie é carregada de 18249 para X e adicionada a 257, movendo-se um caractere para baixo.



O apontador de dados em U é novamente carregado com o endereço inicial da figura de Willie com as pernas abertas em 17814. A rotina **CHARPR** imprime sua metade superior. O apontador de tela em X é incrementado, movendo-se para a linha de baixo. U é carregado com o endereço dos dados para a metade inferior da figura e **CHARPR** é chamada, realizando a impressão.

TERRA FIRME

O valor 132 no acumulador desvia o processador para a rotina que faz Willie pisar novamente em terra firme. Se não houver esse valor em A, o processador vai para o rótulo **MFE** — o que nunca deve ocorrer. Um conteúdo de A menor que 132 indica que o processador ficou executando alguma das rotinas anteriores. Isso acontece se a variável de salto em 18261 — endereço de onde vem o valor de A — ainda não foi incrementada. Temos, assim, um bom dispositivo de prevenção de erros: caso haja um valor incorreto em 18261, o processador salta diretamente para a instrução de limpeza, que reajusta esse endereço com 0.

Se o número em A é 132, o processador continua com esta rotina. Inicialmente, ele vai para a rotina **MFZ**, que apaga a última figura de Willie. Em seguida, a posição de Willie é carregada de 18249 para X. O apontador de tela

em X é guardado na pilha, e somado a 512, passando a apontar para o caractere imediatamente abaixo dos pés de Willie. Os dois bytes são carregados no registrador Y e o apontador de tela original é recuperado da pilha de volta para X.

O conteúdo de Y é comparado com $5555, o valor de um espaço amarelo na tela — ou seja, de um buraco. Se esse valor não estiver presente, o processador vai para o rótulo **MFE**

Se há um buraco ali, nosso personagem precisa de ajuda. X é então incrementado e colocado de volta em 18249, onde está o apontador de posição.

Como você notará, nenhuma verificação é feita quando Willie volta ao solo num nível mais alto. Depois de saltar e avançar, ele sempre aterrissa num nível mais alto.

```
10    org 55009
20    mp cp 129
30    jr nz,mb
40    inc a
50    ld (-5202),a
60    ld hl,(62407)
70    ld de,(-5205)
80    add hl,de
90    ld de,33
100    add hl,de
110    call 74
120    cp 36
130    jp z,54848
140    cp 52
150    jr nz,ma
160    ld hl,(-5205)
170    dec hl
180    ld (-5205),hl
190    ma ld hl,(-5205)
200    ld de,(62407)
210    add hl,de
220    push hl
230    ld de,32
```

AJUSTES FINAIS

O salto agora está completo. Precisamos, no entanto, limpar o contador de salto e a posição que contém o tipo de figura que deve ser impressa. Como essas posições são ajustadas com 0, o processador irá para a primeira parte da rotina de movimentação.

A seção do programa apresentada neste artigo finaliza imprimindo Willie de pé, em sua nova posição. X é carregado com a posição contida em 18249 e U, com a contida em 17774. A rotina **CHARPR** é chamada duas vezes. A última rotina, **MFZ**, é chamada para apagar a figura. X é carregado com a posição de Willie em 18249 e U é carregado com 1536, que equivale a um espaço vazio.

Para testar o programa, digite as seguintes linhas:

```
10 POKE 30000,57:POKE 20847,57:
POKE 20751,57
20 EXEC 19426
30 FOR L=1 TO 8:POKE 18261,129:
FOR J=1 TO 5:EXEC 19902
40 FOR K=1 TO 100:NEXT K,J,L
50 GOTO 50
```

MSX

O programa destinado ao MSX foi dividido em duas partes para facilitar a montagem.

```
240    add hl,de
250    ld a,255
260    push hl
270    call 77
280    pop hl
290    inc hl
300    ld a,255
310    call 77
320    pop hl
330    ld a,5
340    push hl
350    call 77
360    pop hl
370    inc hl
380    ld a,7
390    push hl
400    call 77
410    pop hl
420    ld de,32
430    sbc hl,de
440    ld a,6
450    push hl
460    call 77
470    pop hl
480    dec hl
490    ld a,4
500    call 77
510    ld hl,(-5205)
520    ld de,32
530    sbc hl,de
540    ld (-5205),hl
550    ret
560    mb cp 132
570    jp z,55275
580    inc a
590    ld (-5202),a
600    ld a,(-5203)
610    cp 1
620    jp z,55194
630    ld hl,(-5205)
640    ld de,(62407)
650    add hl,de
660    ld a,255
670    push hl
680    call 77
690    pop hl
700    ld de,32
710    add hl,de
720    ld a,255
730    push hl
740    call 77
750    pop hl
760    inc hl
770    ld a,1
780    push hl
790    call 77
800    pop hl
810    ld de,32
820    sbc hl,de
830    ld a,0
840    call 77
850    ld hl,(-5205)
860    inc hl
870    ld (-5205),hl
880    ld a,1
890    ld (-5203),a
900    ret
910    end
```

Complete agora o programa, digitando as seguintes linhas:

```
10    org 55194
20    mc ld hl,(-5205)
30    ld de,(62407)
40    add hl,de
50    ld a,4
60    push hl
70    call 77
80    pop hl
90    inc hl
100    ld a,6
110    push hl
120    call 77
130    pop hl
140    ld de,32
150    add hl,de
160    ld a,7
170    push hl
180    call 77
190    pop hl
200    dec hl
210    ld a,5
220    call 77
230    ld hl,(-5205)
240    inc hl
250    ld (-5205),hl
260    ld de,(62407)
270    add hl,de
280    ld de,64
290    add hl,de
300    call 74
310    cp 52
320    jr nz,ms
330    ld a,0
340    ld (-5202),a
350    ld a,3
360    ld b,5
370    call 57000
380    ms ld a,0
390    ld (-5203),a
400    ret
410    md ld a,0
420    ld (-5202),a
430    ld hl,(-5205)
440    ld de,(62407)
450    add hl,de
460    push hl
470    ld a,255
480    call 77
490    pop hl
500    dec hl
510    ld a,255
520    push hl
530    call 77
540    pop hl
550    ld de,32
560    add hl,de
570    ld a,255
580    push hl
590    call 77
600    pop hl
610    inc hl
620    ld a,255
630    call 77
640    ld hl,(-5205)
650    ld de,32
660    add hl,de
670    ld (-5205),hl
680    jp 54603
690    ret
700    end
```

Na primeira vez que o processador chama a rotina, o acumulador contém 129. Esse número é armazenado na posição de memória −5202 quando as teclas M e N são pressionadas juntas.

SAINDO DO CHÃO

Antes de mais nada, a rotina verifica se o valor que está no acumulador é 129. O número em A poderia ser também 130, 131 ou 132. Caso um desses números esteja presente, as instruções **cp 129** e **jr nz,mb** desviam o programa para o rótulo **mb**. Porém, quando a rotina é chamada pela primeira vez, o número no acumulador é 129.

Depois dessa checagem, o acumulador é incrementado e carregado de volta em −5202. A seguir, o endereço inicial da Tabela de Nomes da VRAM é carregado em HL. A posição de Willie na tela é transferida de −5205 e −5204 para DE e somada em HL. Adiciona-se o valor 33 nesse par de registros através de DE. Com isso, HL passa a apontar para o endereço na TN ocupado pelo código do padrão que está adiante dos pés de Willie. A rotina 74 da ROM é chamada, devolvendo a A o conteúdo da posição da VRAM apontada por HL — ou seja, ela faz a leitura na VRAM.

O código do padrão que se encontra adiante dos pés de Willie é comparado com 36, o código da língua da cobra. Se esse valor estiver presente, Willie fatalmente morrerá, pois está saltando sobre a língua da cobra e será picado. A instrução **jp z, 54848** manda então o processador para a rotina da morte, apresentada anteriormente com o rótulo **mre**. Se Willie não morreu, o processador compara o código do padrão à frente de seus pés com 52, que é o padrão da encosta. Se o valor não estiver presente, a instrução **jr nz,ma** desvia o programa para a rotina **ma**. Caso contrário, a posição do personagem em −5205 é decrementada.

A rotina **ma** coloca a posição de Willie em HL. DE é carregado com o endereço inicial da TN da VRAM e somado em HL. Esse valor é guardado na pilha. Em seguida, o número 32 é somado em HL através de DE, e o apontador passa a indicar os pés na figura anterior de Willie. O número 255, código do padrão de céu, é carregado em A. O apontador em HL é preservado na pilha e a rotina 77 da ROM é chamada, escrevendo o valor de A no endereço da VRAM apontado por HL. Apagamos, assim, os pés da figura anterior de Willie.

O último valor de HL é recuperado da pilha e incrementado. A é carregado com 255 e a rotina 77 é novamente chamada. Procedemos dessa maneira porque a posição de Willie poderia ter sido decrementada no começo da rotina — a operação nos dá a certeza de que os pés de Willie foram apagados.

O apontador inicial é recuperado da pilha. Em seguida, os padrões 4, 5, 6 e 7 são impressos, formando a figura de Willie com as pernas abertas. Acompanhe o apontador em HL e verifique como isso foi feito. A posição de Willie em −5205 e −5204 é atualizada, já que foi deslocada uma posição acima. Depois, o programa retorna partindo desse ponto da rotina.

Quando a rotina for chamada outra vez, o acumulador conterá 130. O processador pula, então, a primeira parte, indo para o rótulo **mb**.

Essa pequena rotina começa verificando o conteúdo do acumulador. Se ele contiver 132, a rotina **md** é chamada; se contiver 130 ou 131, o processador continua executando **mb**. A primeira coisa que essa rotina faz é incrementar A e carregar o resultado de volta em −5202. Em seguida, a variável da caminhada é carregada no acumulador. Isso indica ao processador qual das figuras de Willie é necessária.

O conteúdo do acumulador é comparado com 1. Se esse valor está presente, o processador vai para o rótulo **mc**, em 55194, e imprime a figura de Willie com as pernas abertas. Se a variável é 0, o processador imprime a figura de Willie com as pernas juntas.

Antes de imprimir qualquer uma das figuras, porém, devemos apagar todos os vestígios da imagem anterior. Para isso, a posição de Willie é transferida de −5205 e −5204 para HL. DE é carregado com o endereço inicial da TN da VRAM e somado em HL. A é carregado com 255, o código do padrão de céu. O apontador HL é guardado na pilha e a rotina 77 é chamada, apagando o quarto superior esquerdo da figura de Willie. O apontador é recuperado da pilha e somado com 32. A é carregado com 255. O apontador é guardado na pilha e a rotina 77 é chamada, apagando o quarto inferior esquerdo da figura. Com isso, apagamos a metade esquerda da antiga figura de Willie. A outra metade desaparece com a impressão da nova figura — desta vez, com as pernas abertas.

O apontador é recuperado da pilha e incrementado. A é carregado com 1, o código de um dos padrões da figura. O apontador é novamente colocado na pilha e a rotina 77 é chamada. Finalizando, o apontador é recuperado da pilha e subtraído de 32, movendo-se um

caractere para cima. A é carregado com 0, o código do outro padrão da figura. A rotina 77 é chamada e Willie é impresso com as pernas juntas.

A posição de Willie é transferida para HL, decrementada e devolvida em −5205 e −5204. O acumulador, carregado com 1, é armazenado em −5203, para que, na próxima vez, a outra figura de Willie seja impressa.

SALTO À DISTÂNCIA

Quando a rotina for chamada de novo, a variável da caminhada conterá 1 e o processador irá diretamente para o rótulo **mc**. Nessa rotina, a figura de Willie com as pernas abertas é impressa da mesma maneira que na rotina **ma**, só que sua posição foi incrementada — ou seja, Willie avança no ar!

Os padrões 4, 5, 6 e 7, que compõem a figura do personagem com as pernas abertas, são colocados na TN (na tela) pela rotina 77, que é chamada quatro vezes. Em seguida, a posição de Willie em −5205 e −5204 é colocada em HL, incrementada e devolvida aos endereços, o que faz a figura avançar mais uma posição.

SUBIDA DA ENCOSTA

Depois disso, devemos verificar se nosso personagem está pisando na encosta ou não. Lembre-se de que HL já contém a posição de Willie. DE é carregado com o endereço inicial da TN e somado em HL. O número 64 também é somado em HL através de DE. O apontador está indicando a posição sob os pés de Willie. A rotina 74 é chamada e faz a leitura da VRAM nessa posição. O valor é comparado com 52, o código da encosta. Se esse valor estiver presente, Willie subiu um lance na montanha. Para interromper o salto, coloca-se 0 em −5202. O número 3 é carregado em A e o 5, em B. Esses parâmetros serão utilizados por outra rotina, chamada na seqüência, que realiza a contagem de pontos; o valor 50 será adicionado no escore. Mas, como ainda não digitamos essa rotina, convém colocar uma instrução **ret** provisória nesse endereço, o que podemos fazer por meio do comando BASIC **POKE 57000.201**.

Se o padrão da encosta não estiver presente, as intruções anteriores são ignoradas e o processador continua a partir do rótulo **ms**.

A é carregado com 0 e armazenado em −5203. Isso é feito independentemente de Willie estar pisando na encosta ou não, e indica que a próxima figura a ser impressa é a do personagem com as pernas juntas.

TERRA FIRME

Quando todas as outras figuras de Willie já tiverem sido impressas na tela, a variável do salto terá o valor 132. Assim, quando a rotina de movimentação for chamada novamente, o processador irá direto para a última rotina, que começa no rótulo **md**.

Em primeiro lugar, **md** ajusta a variável de salto com 0. Esse valor é carregado no acumulador e armazenado em −5202. Em seguida, a figura anterior de Willie com as pernas abertas é apagada — para isso, coloca-se o padrão 255 nas quatro posições, por meio da rotina 77. A posição de Willie é carregada em HL e somada com 32 através do par DE. Willie volta, com isso, a pisar em terra firme. O resultado é armazenado em −5205 e −5204.

Finalmente, a instrução **jp** chama a rotina de movimentação. Essa rotina, além de imprimir a figura de Willie de pé sobre o solo, pesquisará o teclado mais uma vez.

FIGURAS TRIDIMENSIONAIS

Com a técnica ensinada neste artigo, você poderá exibir na tela de seu micro diferentes objetos geométricos sólidos. Trace o perfil da figura escolhida e deixe a máquina fazer o resto por você.

Desenhar objetos em três dimensões, mesmo que simétricos, é uma tarefa difícil — mas não para quem tem um microcomputador. Com o programa apresentado neste artigo, tudo o que você tem a fazer é desenhar o perfil de um dos lados da figura. A máquina completará o trabalho, inclusive preenchendo as faces do sólido com traços que permitirão uma melhor visualização.

A principal função do programa é promover a rotação do perfil que você definiu em torno de um eixo central. Pode-se, portanto, desenhar qualquer figura que possua seções circulares — como vasos, garrafas, candelabros, copos, maçãs, laranjas, enfim, uma enorme variedade de objetos.

Como o programa *roda* a linha original, a figura obtida é denominada *sólido de revolução*. As versões para todos os micros possibilitam que se veja o produto final de qualquer ângulo. As do Spectrum e do TRS-Color realizam uma pequena animação, girando a figura em torno de seu eixo.

ROTAÇÃO DA FIGURA

Para desenhar o perfil do sólido, movimentaremos uma linha na tela, fixando-a quando atingir uma posição considerada satisfatória. Essa técnica, já empregada em outros programas de *INPUT*, permite a visualização de cada uma das etapas do desenho.

Foi estabelecido um limite de vinte retas para a definição do perfil desejado. Na maioria dos casos, onde se utilizam apenas cinco ou seis retas, essa quantidade é mais do que suficiente. Porém, se você quiser traçar curvas, terá que construí-las com várias retas bem pequenas, podendo chegar perto daquele limite.

O ÂNGULO DE VISÃO

Uma grande vantagem do programa é permitir ao observador escolher o ângulo do qual quer ver a figura — não só de cima, de baixo ou de frente, mas de qualquer ângulo entre 0° e 180° (um ângulo de 70°, por exemplo, mostra o objeto como se ele estivesse sobre uma mesa). Você pode mudar esse ângulo quantas vezes quiser e o micro redesenhará o sólido em outra posição.

O programa armazena as coordenadas de todas as linhas. Quando você acaba o desenho, ele faz a rotação de cada linha ao redor do eixo central, de 18° em 18°. Executa em vinte etapas a rotação completa. O ângulo de visão já é levado em conta nesta fase; os círculos aparecem mais achatados à medida que nos aproximamos dos 90°.

Assim que o desenho fica pronto, o computador espera que o usuário pressione a barra de espaços para escolher um outro ângulo.

A ANIMAÇÃO

No Spectrum e também no TRS-Color, você poderá ver o objeto executar a rotação, ao pressionar uma outra tecla. Para isso, o computador desenha as posições intermediárias da rotação e as armazena na memória, recorrendo à técnica de páginas gráficas. Depois que todas as páginas estão feitas, elas são exibidas em seqüência, o que produz o efeito de animação.

UTILIZAÇÃO DO PROGRAMA

Digite o programa e tente traçar algumas figuras sólidas. O processo de desenho consiste em movimentar o cursor para a posição desejada e marcar o pon-

- DESENHOS EM TRÊS DIMENSÕES
- CRIE ALGUMAS FIGURAS
- CONSTRUÇÃO DO PERFIL
- COMO ALTERAR
- O ÂNGULO DE VISÃO
- ROTAÇÃO DA FIGURA
- COMO FUNCIONA O PROGRAMA
- AS TRÊS ROTINAS PRINCIPAIS

to inicial. A partir daí, a reta poderá ser deslocada para qualquer lugar. Quando você estiver satisfeito com sua localização, use uma tecla para fixá-la. Prossiga assim até terminar o perfil da figura. Para indicar ao micro que o desenho está pronto, você deverá pressionar uma outra tecla.

No Spectrum, use as próprias teclas do cursor para mover a linha, **M** para marcar a posição inicial e **<ENTER>** para fixar a reta. Pressione **Q** ao terminar o desenho.

No TRS-Color, as teclas do cursor também direcionam a linha; **<ENTER>** marca o ponto inicial, a barra de espaços fixa a reta e **Y** finaliza a figura.

Para o MSX, use as setas para deslocar a linha, **I** para inicializar o desenho, a barra de espaços para fixar a reta e **F** para finalizar. As teclas **R** e **D** permitem que você desenhe mais rápido ou mais devagar.

No Apple e no TK-2000, use **Z X ; e .** para movimentar a linha. Os outros comandos são iguais aos do MSX.

```
10 PRINT AT 0,10; INK 2;
PAPER 5;"ROTACAO 3D": PAUSE
 100
20 PAUSE 2500
30 CLEAR 30000: PAPER 0: INK
5: CLS : POKE 23658,0
40 GOSUB 2000: BORDER 0
50 PRINT AT 0,0; PAPER 6; INK
0;" MOVER LINHA = TECLAS DE C
URSOR  (CAPSHIFT P/MAIOR VELO
C.)      "
60 PRINT : PRINT  PAPER 6;
INK 0;" FIXAR LINHA = ENTER
        "'" FIXAR CURSOR =
 M     FIM = Q    "
70 INPUT "ANGULO DE OBSERVACA
O (0-180)";i: IF i<0 OR i>180
THEN  GOTO 70
80 PLOT 60,100: DRAW 135,0:
DRAW 0,-49: DRAW -135,0: DRAW
0,49
90 LET is=SIN (i/180*PI)
100 LET bf=0: LET a$=""
110 LET x=128: LET y=51
120 LET xx=x: LET yy=y
130 OVER 1
140 PLOT x,y: DRAW xx-x,yy-y
150 PLOT x,y: DRAW xx-x,yy-y
160 LET z$=INKEY$: IF z$=""
THEN  GOTO 140
170 LET z=CODE z$
180 IF z=13 THEN  GOSUB 500
190 IF z=113 THEN  GOTO 600
195 IF z=109 AND bf<>1 THEN
PLOT 255-x,y: LET x=xx: LET y=
yy: LET bf=1
200 IF z=53 AND xx>128 THEN
LET xx=xx-1
210 IF z=8 AND xx>130 THEN
LET xx=xx-2
220 IF z=55 AND yy<100 THEN
LET yy=yy+1
230 IF z=11 AND yy=98 THEN
LET yy=yy+2
240 IF z=56 AND xx<195 THEN
LET xx=xx+1
250 IF z=9 AND xx<194 THEN
LET xx=xx+2
260 IF z=54 AND yy>52 THEN
LET yy=yy-1
270 IF z=10 AND yy>51 THEN
LET yy=yy-2
280 GOTO 140
500 OVER 0: PLOT x,y: DRAW xx-
x,yy-y
510 PLOT 255-x,y: DRAW x-xx,yy
-y
520 LET x=xx: LET y=yy: LET a$
=a$+CHR$ (x-128)+CHR$ (y-51):
OVER 1: RETURN
600 INK 7: CLS : OVER 0: DIM a
(20,2)
610 FOR a=0 TO 7
620 PRINT AT 21,7; PAPER 2;
BRIGHT 1;"DESENHANDO FIGURA ";
a+1
630 GOSUB 1000
640 IF a<>0 THEN  GOTO 670
650 PRINT AT 19,0; INVERSE 1;"
 <SPACE> P/ ALTERAR ANGULO DE
   OBSERVACAO  -  QUALQUER OUT
RA   TECLA PARA CONTINUAR.
      "
660 LET X$=INKEY$: IF X$=""
THEN  GOTO 650
662 IF X$<>" " THEN  GOTO 670
664 INPUT "DIGITE O NOVO ANGUL
O (0-180)";i
666 IF i<0 OR i>180 THEN  GOTO
662
668 CLS : LET is=SIN (i/180*PI
): GOTO 620
670 GOSUB 1600
680 CLS
690 NEXT a
700 PRINT AT 20,11; PAPER 7;
INK 1;" FIGURA: ";
710 FOR a=128 TO 240 STEP 16
720 POKE 30114,a: RAND USR
30112
730 PRINT AT 20,19;a/16
 740 NEXT a: GOTO 710
1000 FOR b=1 TO LEN a$ STEP 2
1010 LET x=CODE a$(b)
1020 LET y=CODE a$(b+1)
1030 GOSUB 1500
1040 NEXT b
1050 RETURN
1500 FOR c=0 TO 399 STEP 20
1510 LET d=c+(2.25*a): LET py=i
s*y
1515 IF d>=360 THEN  LET d=d-36
0
1520 GOSUB 1530: NEXT c
```

```
1525 DRAW bx-xd,by-yd: RETURN
1530 LET yd=SIN (d/180*PI)*x*CO
S (i/180*PI)
1532 LET xd=COS (d/180*PI)*x
1535 LET xd=128+xd: LET yd=90+y
d+py
1540 IF c=0 THEN  PLOT xd,yd: L
ET bx=xd: LET by=yd
1550 DRAW xd-PEEK 23677,yd-PEEK
 23678
1560 IF b=1 AND bf=1 THEN  GOTO
 1580
1565 IF b=1 THEN  PLOT 128,90:
DRAW xd-128,yd-90: GOTO 1580
1570 PLOT a(1+c/20,1),a(1+c/20,
2): DRAW xd-a(1+c/20,1),yd-a(1+
c/20,2)
1580 LET a(1+c/20,1)=xd
1585 LET a(1+c/20,2)=yd
1590 RETURN
1600 POKE 30102,128+a*16
1610 RAND USR 30100
1620 RETURN
2000 RESTORE 2050
2010 FOR a=0 TO 23
2020 READ b: POKE a+30100,b
2030 NEXT a
2040 RETURN
2050 DATA 17,0,0,33,0,64,1,0,16
,237,176,201
2060 DATA 33,0,0,17,0,64,1,0,16
,237,176,201
```

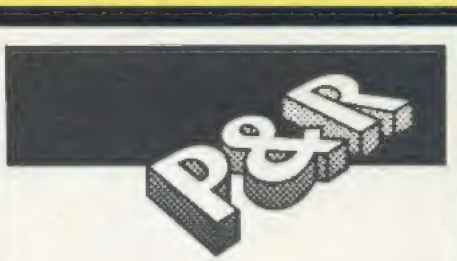
```
10 PMODE 2,1:PCLEAR 2:CLEAR 200
,7679:DIM A(19,1)
20 RD=ATN(1)/45:V=247
30 PCLS 1:SCREEN 1,0
40 BX=128:BY=190:DRAW"BM128,190
":X=BX:Y=BY
50 LINE (31,191)-(255,140),PRES
ET,B
60 LINE (BX,BY)-(X,Y),PRESET
70 IF PEEK(345)=V GOSUB 500
80 IF PEEK(339)=191 THEN M=4 EL
SE M=1
90 LINE (BX,BY)-(X,Y),PSET:IF P
EEK(338)=191 AND A$="" THEN BX=
X:BY=Y:BF=1:GOSUB 500
100 IF PEEK(339)=V AND LEN(A$)>
0 THEN 160
110 IF PEEK(341)=V AND Y-M>141
THEN Y=Y-M
120 IF PEEK(342)=V AND Y+M<191
THEN Y=Y+M
130 IF PEEK(343)=V AND X-M>127
THEN X=X-M
140 IF PEEK(344)=V AND X+M<223
THEN X=X+M
150 GOTO 60
160 GOSUB 2000:SCREEN 1,0:FOR A
=0 TO 6:PCLS1:FOR G=0 TO A:PRES
ET(0,G*2):NEXT
170 GOSUB 1000
180 G$=INKEY$
190 IF A=0 AND G$="" THEN 180
200 IF A=0 AND G$=" " THEN 160
210 NEXT
220 FOR A=0 TO 6:PCOPY 5+A*2 TO
 1:PCOPY 6+A*2 TO 2:NEXT
230 IF INKEY$=" " THEN 160 ELSE
 220
500 LINE(BX,BY)-(X,Y),PRESET
510 LINE(255-BX,BY)-(255-X,Y),P
RESET
520 BX=X:BY=Y:A$=A$+CHR$(X-128)
+CHR$(190-Y):RETURN
1000 FOR B=1 TO LEN(A$) STEP 2
1010 X=ASC(MID$(A$,B,1)):Y=ASC(
MID$(A$,B+1,1))
1020 GOSUB 1500:NEXT
1030 PCOPY 1 TO 5+A*2:PCOPY 2 T
O 6+A*2
1040 RETURN
1500 FOR C=0 TO 360 STEP 20
1510 D=C+20*A/7:PY=IS*Y
1520 GOSUB 1530:NEXT:RETURN
1530 YD=110-SIN(D*RD)*X*COS(I*R
D)-PY:XD=128+COS(D*RD)*X
1540 IF C=0 THEN LINE(XD,YD) (X
D,YD),PRESET:BX=XD:BY=YD
1550 LINE-(XD,YD),PRESET
1560 IF B=1 AND BF=1 THEN 1580
ELSE IF B=1 THEN LINE(128,110)-
(XD,YD),PRESET:GOTO 1580
1570 LINE(A(C/20,0),A(C/20,1))-
(XD,YD),PRESET
1580 A(C/20,0)=XD:A(C/20,1)=YD:
RETURN
2000 CLS:INPUT" ANGULO DE OBSER
VACAO (0-180) ";I
2010 IF I<0 OR I>180 THEN 2000
2020 IS=SIN(I*RD):RETURN
```

```
10 SCREEN 2:DIM A(19,1)
20 RD=ATN(1)/45:M=1
30 LINE (127,66)-(207,126),15,B
F
35 LINE (47,66)-(127,126),14,BF
40 BX=127:BY=126:X=BX:Y=BY:XX=B
X:YY=BY
60 LINE (BX,BY)-(X,Y),8
70 C$=INKEY$:IF C$=""THEN 70
75 IF C$=" " THEN GOSUB 500
80 IF C$="R" THEN M=4
85 IF C$="D" THEN M=1
90 LINE(BX,BY)-(X,Y),15:IF C$="
I" AND A$="" THEN BX=X:BY=Y:BF=
1:GOSUB 500
100 IF C$="F" AND LEN(A$)>0 THE
N 160
110 IF C$=CHR$(30) AND Y-M>66 T
HEN Y=Y-M
120 IF C$=CHR$(31) AND Y+M<126
THEN Y=Y+M
130 IF C$=CHR$(29) AND X-M>127
THEN X=X-M
140 IF C$=CHR$(28)AND X+M<207 T
HEN X=X+M
150 GOTO 60
160 GOSUB 2000
170 COLOR,11,11:SCREEN 2:GOSUB
```

```
1000
190 C$=INKEY$:IF C$="" THEN 190
200 GOTO 160
500 LINE (BX,BY)-(X,Y),1
510 LINE (255-BX,BY)-(255-X,Y),
1
520 BX=X:BY=Y:A$=A$+CHR$(X-126)
+CHR$(127-Y):RETURN
1000 FOR B=1 TO LEN(A$) STEP 2
1010 X=ASC(MID$(A$,B,1)):Y=ASC(
MID$(A$,B+1,1))
1020 GOSUB 1500:NEXT B
1040 RETURN
1500 FOR C=0 TO 399 STEP 20
1510 D=C:PY=IS*(Y*.5)
1520 GOSUB 1530:NEXT C:RETURN
1530 YD=96-SIN(D*RD)*X*COS(I*RD
)-PY:XD=127+COS(D*RD)*X
1540 IF C=0 THEN BX=XD:BY=YD:XX
=XD:YY=YD
1550 LINE(XX,YY)-(XD,YD),1:XX=X
D:YY=YD
1560 IF B=1 AND BF=1 THEN 1580
1565 IF B=1 THEN LINE(80,91)-(X
D,YD),1:GOTO 1580
1570 LINE(A(C/20,0),A(C/20,1))-
(XD,YD),1
1580 XX=XD:YY=YD:A(C/20,0)=XD:A
(C/20,1)=YD:RETURN
2000 SCREEN 0:KEY OFF:COLOR 15,
4,4:LOCATE 8,8 :INPUT"Ângulo de
 visão:";I
2010 IF I<0 OR I>180 THEN 2000
2020 IS=SIN(I*RD):RETURN
```

```
10  HOME : HGR : DIM A(19,1)
20 RD =  ATN (1) / 45:M = 1
30  HCOLOR= 6: HPLOT 140,90 TO
220,90 TO 220,149
40 BX = 140:BY = 149:X = BX:Y =
 BY:XX = BX:YY = BY
50  VTAB (21): PRINT "COMANDOS:
 <I>NICIO-<F>IM
52  VTAB (22): PRINT "VELOCIDAD
E DO RISCO: <R>APIDO-<D>EVAGAR"

54  VTAB (23): PRINT "BARRA DE
ESPACO: FIXA A RETA"
56  VTAB (24): PRINT "DIRECAO:
<Z>,<X>,<;>,<.>";
60  HCOLOR= 3: HPLOT BX,BY TO X
,Y
70  GET C$: IF C$ = " " THEN  G
OSUB 500
80  IF C$ = "R" THEN M = 4
85  IF C$ = "D" THEN M = 1
90  HCOLOR= 0: HPLOT BX,BY TO X
,Y: IF C$ = "I" AND A$ = "" THE
N BX = X:BY = Y:BF = 1: GOSUB 5
00
100  IF C$ = "F" AND  LEN (A$)
> 0 THEN 160
110  IF C$ = ";" AND Y - M > 90
 THEN Y = Y - M
120  IF C$ = "." AND Y + M < 15
0 THEN Y = Y + M
130  IF C$ = "Z" AND X - M > 14
0 THEN X = X - M
140  IF C$ = "X" AND X + M < 22
0 THEN X = X + M
150  GOTO 60
160  HGR : GOSUB 2000
170  GOSUB 1000
190  GET C$: IF C$ = " " THEN 1
60
195  GOTO 190
500  HCOLOR= 3: HPLOT BX,BY TO
X,Y
510  HPLOT 281 - BX,BY TO 281 -
 X,Y
520 BX = X:BY = Y:A$ = A$ +  CH
R$ (X - 140) +  CHR$ (149 - Y):
 RETURN
1000  FOR B = 1 TO  LEN (A$) ST
EP 2
1010 X =  ASC ( MID$ (A$,B,1)):
Y =  ASC ( MID$ (A$,B + 1,1))
1020  GOSUB 1500: NEXT B
1040  RETURN
1500  FOR C = 0 TO 399 STEP 20
1510 D = C:PY = IS * (Y * .5)
1520  GOSUB 1530: NEXT C: RETUR
N
1530 YD = 80 -  SIN (D * RD) *
X *  COS (I * RD) - PY:XD = 140
 +  COS (D * RD) * X
```



É possível realizar em um micro animações gráficas em três dimensões como as que aparecem na TV?

Os grafismos animados em três dimensões que aparecem em vinhetas de televisão, e nos filmes produzidos por computador (*TRON*, por exemplo), devem sua espantosa qualidade e realismo às máquinas que os elaboram — geralmente, supercomputadores do tipo Cray II, com capacidade de executar sessenta milhões de operações em ponto flutuante por segundo.

A técnica utilizada, entretanto, não difere da que explicamos neste artigo para gerar sólidos de revolução. Superfícies complexas são montadas por milhares de "tijolos" regulares, depois, "alisados" por algoritmos especiais, o que lhes dá um aspecto contínuo. Outros programas pintam as superfícies com cores selecionadas, adicionam brilho e sombra etc., tornando o resultado o mais realista possível.

Os micros podem, teoricamente, executar quase tudo o que um computador de grande porte faz. Porém, têm telas com menor resolução gráfica, menor número de cores, enfim, não contam com os mesmos recursos. Não se pode esperar, portanto, que produzam efeitos de igual qualidade.

A figura tridimensional é construída a partir de um simples perfil.

O microcomputador elabora o desenho executando a rotação de cada reta.

O produto final — o *sólido de revolução* — é exibido em três dimensões.

```
1540  IF C = 0 THEN BX = XD:BY
= YD:XX = XD:YY = YD
1550  HCOLOR= 3: HPLOT XX,YY TO
 XD,YD:XX = XD:YY = YD
1560  IF B = 1 AND BF = 1 THEN
1580
1565  IF B = 1 THEN  HPLOT 80,9
1 TO XD,YD: GOTO 1580
1570  HPLOT A(C / 20,0),A(C / 2
0,1) TO XD,YD
1580 XX = XD:YY = YD:A(C / 20,0
) = XD:A(C / 20,1) = YD: RETURN
```

```
2000  HOME : VTAB (23): INPUT "
ANGULO DE VISAO (0-180): ";I
2010  IF I < 0 OR I > 180 THEN
2000
2020 IS =  SIN (I * RD): RETURN
```

COMO FUNCIONA

Todos os programas funcionam de modo muito semelhante. As partes mais importantes são as rotinas que permitem o traçado do perfil, criam o sólido de revolução e fazem-no girar.

INTRODUÇÃO DO PERFIL

A rotina de desenho começa na linha 60 (linha 140, no micro Spectrum). Ela simplesmente lê as teclas que foram pressionadas e atualiza o valor das coordenadas da linha auxiliar. O desenho definitivo é feito pela rotina da linha 500, que também traça o perfil esquerdo da figura.

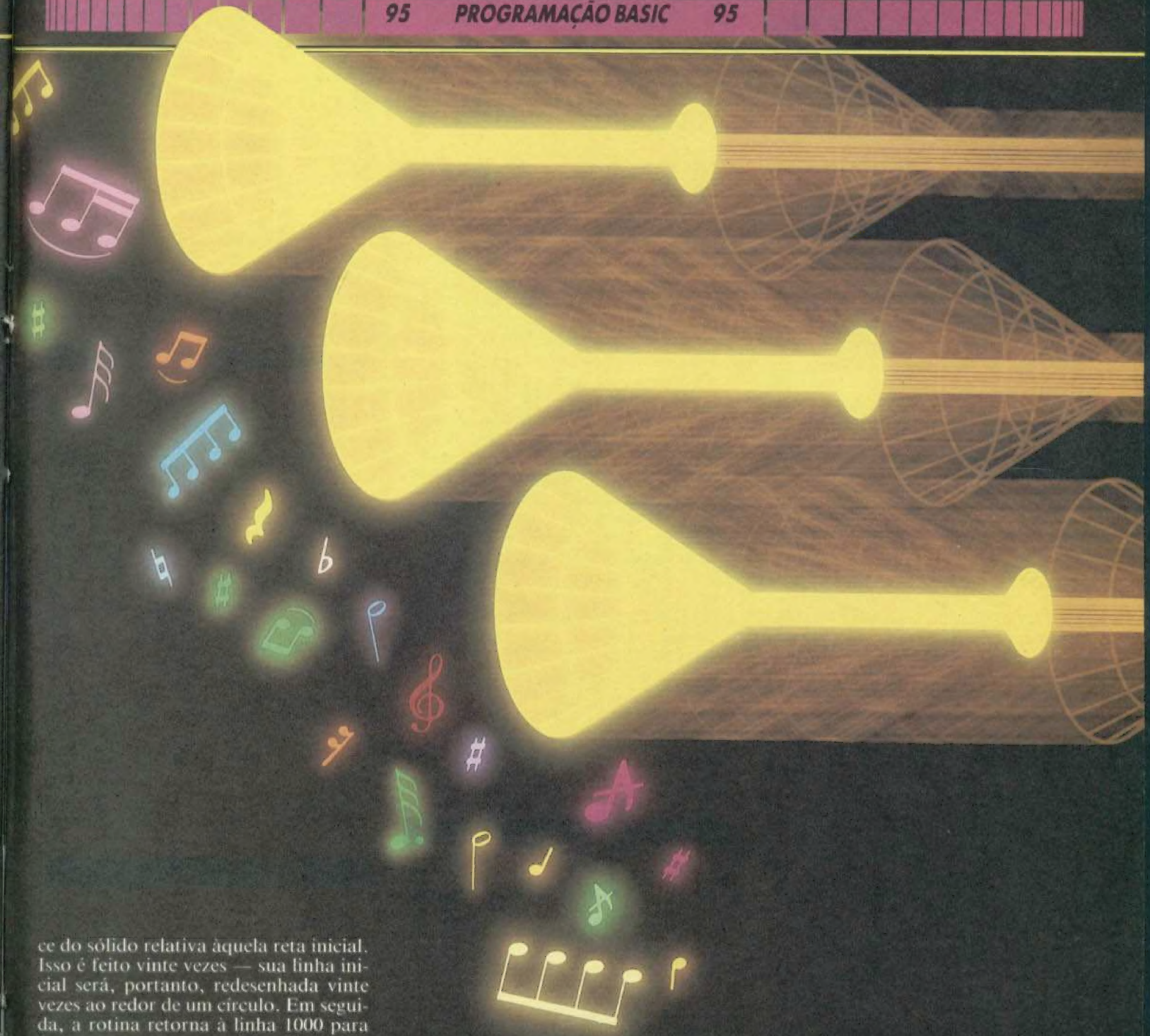
Cada vez que se fixa uma linha, as coordenadas de sua extremidade são armazenadas. Primeiro, subtrai-se um número de cada coordenada para obter uma nova coordenada, relacionada ao ponto de partida do desenho. Em seguida, esses números são convertidos em caracteres e adicionados ao final de uma variável alfanumérica (**A$** ou **a$**).

A rotina da linha 2000 (70, no Spectrum) permite que se defina o ângulo de visão. Depois, o programa é desviado para a linha 1000, que cria e desenha o sólido em três dimensões.

TERCEIRA DIMENSÃO

Essa rotina utiliza as coordenadas previamente armazenadas e o ângulo de visão que você definiu para transformar o perfil em um sólido de revolução. A técnica matemática usada por esta rotina é a mesma para todos os computadores, embora os comandos gráficos sejam diferentes.

Na linha 1000, um laço controlado pela variável B seleciona cada uma das suas linhas, tomando as coordenadas finais da variável **A$**. O programa salta, então, para a rotina da linha 1500. Essa rotina desloca as coordenadas finais de acordo com o ângulo requerido e traça as novas retas que irão compor a fa-



ce do sólido relativa àquela reta inicial. Isso é feito vinte vezes — sua linha inicial será, portanto, redesenhada vinte vezes ao redor de um círculo. Em seguida, a rotina retorna à linha 1000 para fazer o mesmo com a próxima linha. Ao executar o programa, você poderá observar essas etapas da elaboração do sólido.

Como você já deve ter observado, o processo empregado para desenhar a primeira linha da figura é um pouco diferente do utilizado para as demais, já que esta linha possui duas coordenadas, a inicial e a final.

As linhas 1560 e 1565 verificam se a reta a ser trabalhada é a inicial. Se **B** = 1, a reta é realmente a primeira de todas; se **BF** = 1, ela não deve ser traçada, pois corresponde apenas ao deslocamento para a definição do ponto inicial. Se houve um deslocamento, o programa armazenará as coordenadas do ponto inicial na variável alfanumérica. Essas coordenadas serão o ponto de partida para a próxima linha. Caso o primeiro movimento não tenha sido um deslocamento, o ponto inicial será o centro, a posição inicial do cursor; o ponto fixado, por sua vez, será a coordenada final da primeira reta.

Através deste processo, cada reta do perfil será "multiplicada" por vinte e todas elas terão suas extremidades ligadas por um círculo, gerando o efeito de três dimensões.

A ROTAÇÃO

O Spectrum e o TRS-Color, como dissemos, ainda oferecem o artifício de animar a figura na tela, ao toque de uma tecla. O número de figuras (**A**) é controlado pelo laço das linhas 610 a 690, no Spectrum, e da linha 220 no TRS-Color. A variável **A** controla também a rotação parcial de cada figura, por meio de uma modificação no ângulo de visão (linha 1510). Finalmente, os quadros são exibidos em seqüência, criando o efeito de animação.

VIDEOTEXTO E MICROCOMPUTADORES

- O QUE É O VIDEOTEXTO
- COMO FUNCIONA
- INFORMAÇÕES E SERVIÇOS
- CONEXÃO AO VIDEOTEXTO
- COMPATIBILIDADE

Use o micro para ter acesso a um sofisticado sistema de intercâmbio de informações: o videotexto. Ele colocará à sua disposição uma variada gama de serviços.

Utilizado isoladamente, um microcomputador doméstico é capaz de desempenhar uma grande variedade de tarefas. Estas se multiplicam, porém, quando ele passa a integrar uma *rede de computadores* — o que, nos últimos anos, se tornou possível no Brasil.

No artigo da página 561, vimos como a tecnologia dos *modems* (modulador-demodulador) permite que se recorra à rede telefônica normal para envio e recepção de mensagens, textos, dados, programas e gráficos em forma digital. Um computador pode, assim, "conversar" com outro de uma maneira barata e relativamente rápida.

Existem diferentes tipos de rede de computadores que empregam o telefone e o modem como meios de comunicação. As redes de área local (LAN) ou ampla (WAN), compostas apenas por microcomputadores (como os quadros de avisos, ou *bulletins boards*), serão examinadas em outro artigo. Aqui, trataremos das redes em que o microcomputador é usado como terminal de um computador maior.

COMO FUNCIONA

O videotexto é um exemplo desse tipo de rede. O sistema estabelece o contato do usuário com um computador de grande porte, por meio da rede telefônica normal. O microcomputador não é indispensável à rede: o usuário precisa ter apenas um terminal de videotexto. Este é um pequeno aparelho dotado de microprocessador e teclado portátil, que usa como saída de vídeo um aparelho de TV doméstico.

O software operacional do videotexto permite não só a implementação de uma estrutura própria de dados, bem como sua consulta pelo usuário. As informações armazenadas no computador central do sistema videotexto são organizadas em *páginas*. Cada página corresponde a uma ou mais telas de vídeo contendo texto e ilustrações, ambos em cores. Para sua composição, utiliza-se um conjunto de caracteres gráficos típicos de videotexto.

As páginas são organizadas segundo uma estrutura em forma de árvore. O usuário pode "explorar" à vontade essa árvore, chamando à tela — por meio de seleções efetuadas pelo teclado — as diversas páginas que compõem um determinado conjunto. As opções feitas são enviadas ao computador central, que então encaminha a página solicitada para o terminal de videotexto, por intermédio da linha telefônica.

O uso do videotexto é aberto a assinantes — ou seja, o usuário que desejar ter acesso ao mesmo deve pagar uma assinatura mensal, debitada na conta telefônica normal, além de uma taxa de utilização, que varia proporcionalmente ao tempo que o terminal permanece conectado ao sistema.

Os dados armazenados no computador central são fornecidos por empresas, órgãos governamentais, jornais, revistas e outras fontes, que mantêm suas próprias páginas de informação, atualizando-as freqüentemente.

O videotexto oferece uma enorme variedade de serviços informativos e recreativos: notícias do dia, indicadores econômicos e cotações das bolsas, programação de cinemas e teatros, publicações, transportes aéreos, turismo, informações científicas e médicas, horóscopo, números da lista telefônica, produtos à venda etc. Como o sistema é interativo, inclui ainda serviços em que a participação do usuário é mais intensa, tais como: videogames, troca de mensagens pessoais, anúncios classificados, reserva de passagens aéreas, solicitação de catálogos, concursos, telecompras etc.

O sistema é fácil de usar, fornecendo ao usuário todas as instruções necessárias e os menus para seleção.

CONEXÃO A UM MICRO

Como já dissemos anteriormente, mesmo aqueles que não possuem um computador doméstico podem se associar ao sistema videotexto utilizando um terminal alugado pela própria companhia telefônica local por um valor módico (não é necessário comprá-lo).

Entretanto, a conexão de um micro ao sistema videotexto traz algumas vantagens ao usuário — entre elas o acesso a alguns serviços de fornecimento gratuito de software, que pode ser copiado com grande facilidade em fita cassete ou disco. Dispondo de um computador, o usuário tem ainda a possibilidade de listar em impressora as informações fornecidas pelo sistema, para exame posterior.

Para conectar o seu micro ao sistema videotexto, basta ter:

- uma porta serial assíncrona, do tipo RS-232C (se seu computador não dispõe de uma, você precisará comprar uma placa de expansão);

- um modem de 1200 bips por 75 (existem tipos especiais para videotexto, alguns deles disponíveis também como placas internas de expansão);

- um programa de emulação de terminal de videotexto, que deve ser carregado previamente na memória do seu micro, quando quiser acessar o serviço.

Já existem "pacotes" para acesso ao videotexto, contendo os componentes mencionados, para praticamente todos os tipos de microcomputadores nacionais. Algumas companhias telefônicas alugam ou vendem *kits* de acesso para os modelos mais populares, como o Apple II e seus compatíveis. Mesmo micros bem simples, como o TK-90X, podem ser conectados ao videotexto (não é preciso ter uma unidade de disquete).

Finalmente, para se beneficiar do sistema, será necessário que você se inscreva como assinante do videotexto junto à companhia telefônica. Consulte-a para saber se o serviço está disponível em sua cidade e se seu micro foi credenciado para acesso ao videotexto. Em caso de resposta afirmativa, você poderá obter também a informação sobre onde conseguir o *kit* de acesso, para aluguel ou compra.

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR LINHA

LINHA	FABRICANTE	MODELO
Apple II +	Appletronica	Thor 2010
Apple II +	CCE	MC-4000 Exato
Apple II +	CPA	Absolutus
Apple II +	CPA	Polaris
Apple II +	Digitus	DGT-AP
Apple II +	Dismac	D-8100
Apple II +	ENIAC	ENIAC II
Apple II +	Franklin	Franklin
Apple II +	Houston	Houston AP
Apple II +	Magnex	DM II
Apple II +	Maxitronica	MX-2001
Apple II +	Maxitronica	MX-48
Apple II +	Maxitronica	MX-64
Apple II +	Maxitronica	Maxitronic I
Apple II +	Microcraft	Craf II Plus
Apple II +	Milmar	Apple II Plus
Apple II +	Milmar	Apple Master
Apple II +	Milmar	Apple Senior
Apple II +	Omega	MC-400
Apple II +	Polymax	Maxxi
Apple II +	Polymax	Poly Plus
Apple II +	Spectrum	Microengenho I
Apple II +	Spectrum	Spectrum ed
Apple II +	Suporte	Venus II
Apple II +	Sycomig	SIC I
Apple II +	Unitron	AP II
Apple II +	Victor do Brasil	Elppa II Plus
Apple II +	Victor do Brasil	Elppa Jr.
Apple IIe	Microcraft	Craft IIe
Apple IIe	Microdigital	TK-3000 IIe
Apple IIe	Spectrum	Microengenho II
MSX	Gradiente	Expert GPC-1
MSX	Sharp	Hotbit HB-8000
Sinclair Spectrum	Microdigital	TK-90X
Sinclair Spectrum	Timex	Timex 2000
Sinclair ZX-81	Apply	Apply 300
Sinclair ZX-81	Engebras	AS-1000
Sinclair ZX-81	Filcres	NEZ-8000
Sinclair ZX-81	Microdigital	TK-82C
Sinclair ZX-81	Microdigital	TK-83
Sinclair ZX-81	Microdigital	TK-85
Sinclair ZX-81	Prologica	CP-200
Sinclair ZX-81	Ritas	Ringo R-470
Sinclair ZX-81	Timex	Timex 1000
Sinclair ZX-81	Timex	Timex 1500
TRS-80 Mod. I	Dismac	D-8000
TRS-80 Mod. I	Dismac	D-8001/2
TRS-80 Mod. I	LNW	LNW-80
TRS-80 Mod. I	Video Genie	Video Genie I
TRS-80 Mod.III	Digitus	DGT-100
TRS-80 Mod.III	Digitus	DGT-1000
TRS-80 Mod.III	Kemitron	Naja 800
TRS-80 Mod.III	Prologica	CP-300
TRS-80 Mod.III	Prologica	CP-500
TRS-80 Mod.III	Sysdata	Sysdata III
TRS-80 Mod.III	Sysdata	Sysdata Jr.
TRS-80 Mod.IV	Multix	MX-Compacto
TRS-80 Mod.IV	Sysdata	Sysdata IV
TRS-Color	Codimex	CS-6508
TRS-Color	Dynacom	MX-1600
TRS-Color	LZ	Color 64
TRS-Color	Microdigital	TKS-800
TRS-Color	Prologica	CP-400

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR NOME

FABRICANTE	MODELO	PAÍS	LINHA
Appletronica	Thor 2010	Brasil	Apple II +
Apply	Apply 300	Brasil	Sinclair ZX-81
CCE	MC-4000 Exato	Brasil	Apple II +
CPA	Absolutus	Brasil	Apple II +
CPA	Polaris	Brasil	Apple II +
Codimex	CS-6508	Brasil	TRS-Color
Digitus	DGT-100	Brasil	TRS-80 Mod.III
Digitus	DGT-1000	Brasil	TRS-80 Mod.III
Digitus	DGT-AP	Brasil	Apple II +
Dismac	D-8000	Brasil	TRS-80 Mod. I
Dismac	D-8001/2	Brasil	TRS-80 Mod. I
Dismac	D-8100	Brasil	Apple II +
Dynacom	MX-1600	Brasil	TRS-Color
ENIAC	ENIAC II	Brasil	Apple II +
Engebras	AS-1000	Brasil	Sinclair ZX-81
Filcres	NEZ-8000	Brasil	Sinclair ZX-81
Franklin	Franklin	USA	Apple II +
Gradiente	Expert GPC1	Brasil	MSX
Houston	Houston AP	Brasil	Apple II +
Kemitron	Naja 800	Brasil	TRS-80 Mod.III
LNW	LNW-80	USA	TRS-80 Mod. I
LZ	Color 64	Brasil	TRS-Color
Magnex	DM II	Brasil	Apple II +
Maxitronica	MX-2001	Brasil	Apple II +
Maxitronica	MX-48	Brasil	Apple II +
Maxitronica	MX-64	Brasil	Apple II +
Maxitronica	Maxitronic I	Brasil	Apple II +
Microcraft	Craft II Plus	Brasil	Apple II +
Microcraft	Caft IIe	Brasil	Apple IIe
Microdigital	TK-3000 IIe	Brasil	Apple IIe
Microdigital	TK-82C	Brasil	Sinclair ZX-81
Microdigital	TK-83	Brasil	Sinclair ZX-81
Microdigital	TK-85	Brasil	Sinclair ZX-81
Microdigital	TK-90X	Brasil	Sinclair Spectrum
Microdigital	TKS-800	Brasil	TRS-Color
Milmar	Apple II Plus	Brasil	Apple II +
Milmar	Apple Master	Brasil	Apple II +
Milmar	Apple Senior	Brasil	Apple II +
Multix	MX-Compacto	Brasil	TRS-80 Mod.IV
Omega	MC-400	Brasil	Apple II +
Polymax	Maxxi	Brasil	Apple II +
Polymax	Poly Plus	Brasil	Apple II +
Prologica	CP-200	Brasil	Sinclair ZX-81
Prologica	CP-300	Brasil	TRS-80 Mod.III
Prologica	CP-400	Brasil	TRS-Color
Prologica	CP-500	Brasil	TRS-80 Mod.III
Ritas	Ringo R-470	Brasil	Sinclair ZX-81
Sharp	Hotbit HB-8000	Brasil	MSX
Spectrum	Microengenho I	Brasil	Apple II +
Spectrum	Microengenho II	Brasil	Apple IIe
Spectrum	Spectrum ed	Brasil	Apple II +
Suporte	Venus II	Brasil	Apple II +
Sycomig	SIC I	Brasil	Apple II +
Sysdata	Sysdata III	Brasil	TRS-80 Mod.III
Sysdata	Sysdata IV	Brasil	TRS-80 Mod.IV
Sysdata	Sysdata Jr.	Brasil	TRS-80 Mod.III
Timex	Timex 1000	USA	Sinclair ZX-81
Timex	Timex 1500	USA	Sinclair ZX-81
Timex	Timex 2000	USA	Sinclair Spectrum
Unitron	AP II	Brasil	Apple II +
Victor do Brasil	Elppa II Plus	Brasil	Apple II +
Victor do Brasil	Elppa Jr.	Brasil	Apple II +
Video Genie	Video Genie I	USA	TRS-80 Mod. I

UM LOGOTIPO PARA CADA MODELO DE COMPUTADOR

INPUT foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Sinclair ZX-81

TRS-80

TK-2000

MSX

Spectrum

TRS-Color

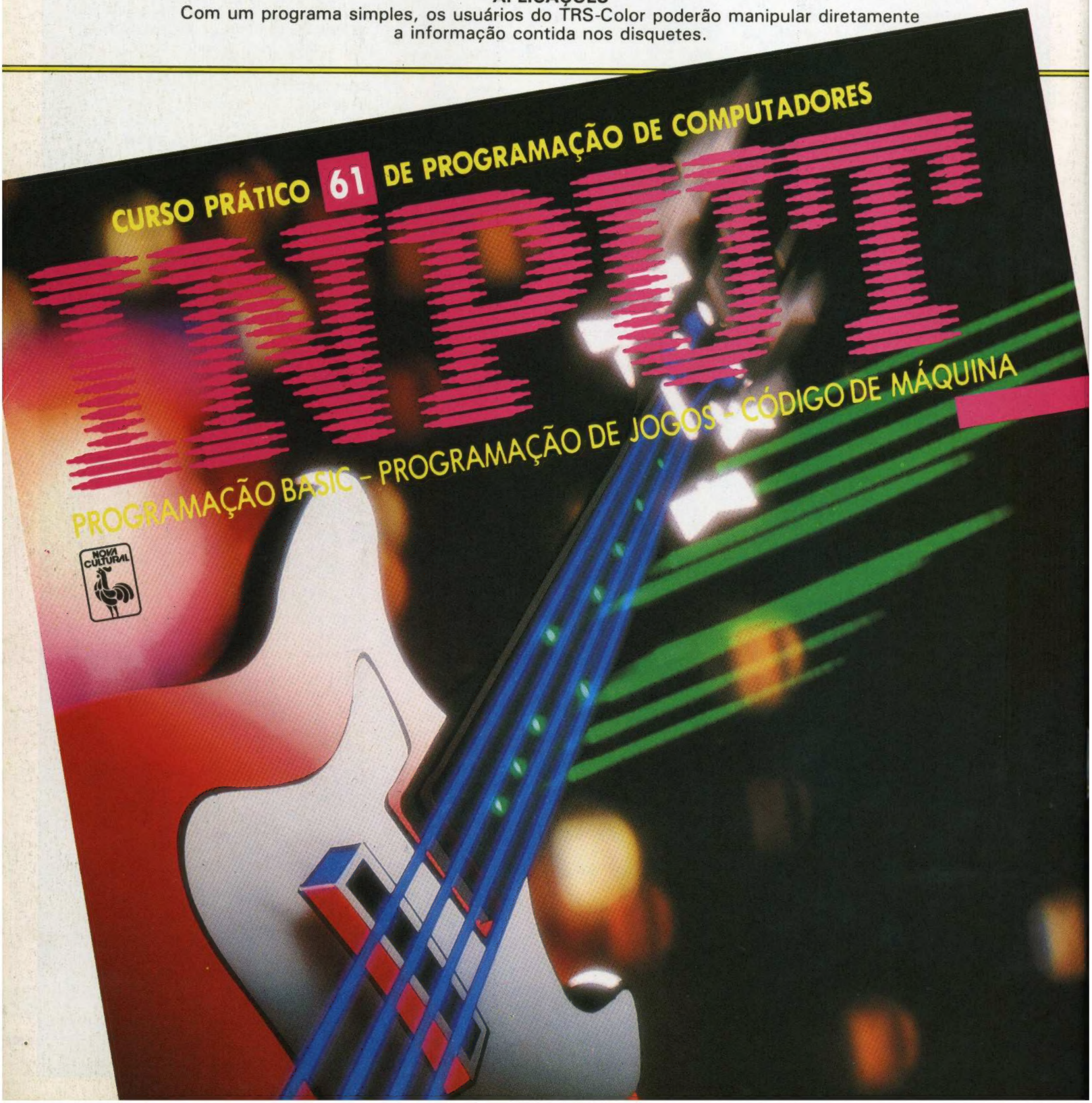
Apple II

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

NO PRÓXIMO NÚMERO

PROGRAMAÇÃO BASIC
Amplie a capacidade musical de seu micro reduzindo os dados necessários à execução de melodias.

CÓDIGO DE MÁQUINA
Avalanche: a última morte de Willie. Funeral. Descida ao inferno. Placar.

PROGRAMAÇÃO BASIC
Veja como usar pequenas rotinas para remover espaços em branco e converter maiúsculas em minúsculas e vice-versa.

APLICAÇÕES
Com um programa simples, os usuários do TRS-Color poderão manipular diretamente a informação contida nos disquetes.